Izdaje BIGZ OOUR „Duga"

računari

specijalno izdanje časopisa „galaksija"

novembar 1985.
izlazi jedampút
mesečno
cena 250 dinara

9



ekskluzivno
razglednica iz londona
razglednica iz njujorka

makazama po rom-u
gena iz rom-a

programski jezici
k… …og komala

računari u izlogu
Zbogom… …a

hakerski manifest
prvi programeri … govore paskal

Izlazi jednom mesečno ✓ Izdaje BIGZ OOUR „Duga“

Cena 250 dinara / novembar 1985

9

računari

Specijalno izdanje časopisa „Galaksija“

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga“
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Debrosav Petrović

Direktor OOUR „Duga“
Bratoljub Babić
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanja
Jova Regasek

Tehnički urednik
Mirko Fegev

Redakcija časopisa „Galaksija“
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik

Stručni saradnici
Dejan Ristanović
Nevenka Spalević
Anđelko Zgorelec
Mihajlo Telečić

Autori tekstova
Aleksandar Demel
Branko Djaković
Mihajlo Karapandžić
Vladimir Kostić
Saša D. Kovačević
Vladimir Krstonošić
Miroslav Mihajlović
Radomir A. Mihajlović
Ivan Nador
Bogdan Petrović
Petar Putnik
Dejan Ristanović
Duško Savić
Jelena Rupnik
Joven Skuljan
Zoran Životić
Zvonimir Vistrička

Tehnička saradnja
Ljubiša Milovanović
Ljev Rjadčenko

Prevodioci
Esad Jakupović
Ksenija Fjedić-Lašedinski
Domagoj Bačić

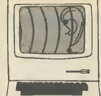
Izdavački savet „Galaksije“
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Mastać, Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17

Žiro-račun kod SDK 60802-603-2463
Devizni račun kod Beobanke
60811-620-6-92701-999-01096
Za inostranstvo cena dvostruka (400 D, 2.50 US$, 6,30 DM, 45 Sch, 5,30 Sfrs, 20 FFRS)

Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/73-03 i, Službeni glasnik broj 28/72, ovo izdanje oslobođeno je poreza na promet

sadržaj



šta ima novo

Računari u obrazovanju

Ne lipši magare do zelene trave

Nedavno je u okviru međunarodnog Sajma knjiga u Beogradu organizovano savetovanje o primeni računara u obrazovanju Tom prilikom je između ostalog održan i razgovor oko okruglog stola na temu „Iskustva republika i pokrajina u primeni informatike i računarstva u obrazovanju'' Učesnici ove diskusije bili su uglavnom pred-



stavnici republičkih i pokrajinskih Zavoda za unapređivanje obrazovanja i vaspitanja

Predstavnik SR Srbije izneo je neke rezultate eksperimentalnog uvođenje nastave informatike i računarstva u osnovno obrazovanje Prošle godine je u eksperimentu učestvovalo 12 osnovnih škola sa oko 280 učenika Rezultati ovog istraživanja pokazali su da su učenici postigli veoma dobar uspeh iz računarskih predmeta gotovo 50% imalo je na kraju godine odlične ili vrlodobre ocene Pokazalo se takođe da su osnovci zainteresovani za nastavu ova vrsta što se moglo i očekivati to je jedini predmet sa čijih časova učenici ne beže, već ostaju i duže nego što je predviđeno

Manje je poznato međutim, da je školama u SR Srbiji upućena molba da se uzdrže od kupovine računara dok nadležni ne donesu odluku koji bi to računar trebalo da bude ili bar koje bi uslove trebalo da zadovoljava Zasad je taj problem donekle rešen u SR Hrvatskoj, a u SR Sloveniji već se zna da računare odgovarajućih karakteristika proizvodi upravo Iskra-Delta u saradnji sa Intertrejdom Sva sreća pa se škole nisu baš previše obazirale na ova upozorenja u stilu „ne lipši magare ' tako da je u aprilu ove godine prebrojano oko 450 računara u osnovnim i srednjim školama na teritoriji SR Srbije od čega najviše ima „galeksija'', spektruma'' „lola 8' i komodora 64''

I u SR Hrvatskoj u toku je uvođenje računarskih sadržaja u osnovno i srednje obrazovanje Istovremeno se radi i na osposobljavanju nastavnika za korišćenje računara i didaktičko-metodoško osmišljavanje rada sa kompjuterom'' U toku je i opremanje škola računarima u skladu sa nedavno usvojenim konceptom kompjuterizacije obrazovanja, u svakoj školi uskoro bi trebalo da se nađe po 15 računara

U SAP Vojvodini u toku je akcija „1000 računara u škola'' Lepo zvuči, ali ne preterano, pogotovo kad se ima u vidu da se radi o 469 što osnovnih što srednjih škola Pored toga u svakoj od njih planira se osnivanja elektronsko-računarske učionice za narednih desetak godina (?)

U SR Crnoj Gori planovi su nešto skromniji — na sto učenika predviđen je jedan računar a u većim školama i printer Slično je i u drugim republikama i pokrajinama, gde su pored akcija opremanja škola računarima u toku i izrada nastavnih planova uvođenja informatike i računarstva u obrazovanja

Na kraju ovog razgovora došlo se do zaključka da je stanje u svim republikama i pokrajinama otprilike ujednačeno i da je sada pravi trenutak za rešavanje problema standardizacije školskih računara i nastavnih programa Pitanje standardizacije ne bi trebalo da predstavlja neki veći problem jer kao što je na kraju diskusija primećeno ako se u svim republikama i pokrajinama Prva pomoć i Opštenarodna odbrana i DSZ uče iz istih udžbenika i uz pomoć istih nastavnih sredstava zašto to ne bi bio slučaj i sa računarskim predmetima?

Jelena Rupnik

Sve se glasnije šuška o Sinklerovom QLu II koji bi trebalo da ima 256K ili 512K memorije i Psionov softver ugrađen u ROM Živi bili pa videli

---

Iznenađenja od Tramiela Na velikom PCW sajmu je predstavio svoj 260 ST sa ugrađenim disk drajvom od 3 5 inča Verovatno to nije jedino iznenađenje iz Tramielovog rukava

---

Još jedna klasika je dobila verziju od 128K To je BBC+ za koji se priča da je katastrofalno loše prošao u prodaji Novi BBC od 128K se prodaje po istoj ceni po kojoj i stari od 64K a to je 499 funti Još uvek preskupo

---

Još jedno govorkanje za novi QL Šuška se da bi mogao imati disk drajv umesto dva mikrodrajva Sinkleru je izgleda došlo iz RAM-a u glavu

---

RAM čipovima svetlosnim penma interfejsima i sličnim hardverskim sitnicama rapidno opada cena Kriza?

Krvni neprijatelji u simbiozi

Oni koji nikako nisu mogli da se odluče za jednog od dva suparnička računarska mamuta više ne moraju da biraju između Epla i IBM-a ako im se sviđaju i makintoš i PC Firma Dajna Komunikejšns (Dayna Communications) proizvodi dodatnu opremu koja običnog meka ' pretvara u mek-čarli'' (Mac-Charlie) mašinu koja je brža komplekarnja-pametnija i — šta sve ne Naravno to nije njena glavna osobina — ono što je bitno je da je mek-čarli kompatibilan sa Ibm-PC-em ' Tako su pomirena dva suprotna a veoma cenjena pola Dobijena je kombinacija makintošove humanosti i softverske prilagođenosti korisniku i enormna programska podrška koju PC ima



Sastavljenje je veoma jednostavno , Mek-čarli ' ima tastaturu u koju se ubacuje makintoševa a telo računara se postavlja na specijalno postolje u kome su sva proširenja i adaptari Ovako napravljen mutant može da se uključi u bilo koju IBM-ovu mrežu a može čak i da se priključi na velike IBM-ova mašine Cenu ovog dodatka nećemo saopštiti da vam ne bi pokvarili dobar utisak ove vesti

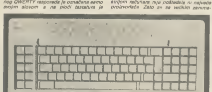
Ploter za narod

Epson je izbacio na tržišta svoj novi četvorobojni ploter pod oznakom HI-80 Ovaj model bi trebalo, po zamisli Epsona da približi ovu retku i egzotičnu napravu i ljudima koji ne plivaju u parama

HI-80 spada u klasu sasvim kvalitetnih plotera To se moglo i očekivati s obzirom na ime proizvođača Spolja izgleda skoro kao printer Na raspolaganju su tri vrate pera sa po deset boja, dok ploter jednovremeno koristi četiri boje Papir se pokreće napred-nazad pomoću mehanizma koji ostavlja skoro nevidljivu rupicu a pera se kreću levo-desno Ploter prima papir veličine do formata A4 Ima tri moda rada, od kojih jedan predstavlja simulaciju printera Preciznost „šava'' pri promeni pera je 0,3 mm Ploter kodati standardni paralelni interfejs Kako Epson namerava da ovu mašinu približi malo širim masama? Ploter košta samo 599 dolara To je upadljivo niža cena od cena ostalih Epsonovih i tuđih kvalitetnijih plotera

Saga za „spektrum"

Za sve one kojima su dozlogrdile gumice na „dugi" Saga je izbacila na tržište tastaturu koja njihov „spektrum" pretvara u nešto što bar liči na pravi računar „Saga 3 Elite" je profesionalna tastatura za „spektrum" koja sadrži sva ona poodišanja koja ste mogli da priželjkujete kod jednog starog „spektruma" Svaka tipka iz standardnog QWERTY rasporeda je označena samo svojim slovom a na ploči tastatura je



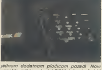
odštampana shema ostalih funkcija Numeričke tipke su izdvojene na desnoj strani, kao i nekoliko tipki sa specifičnim funkcijama kao što su Save, Load Clear Copy i slične Naravno postoje i mali problemi pošto je tastatura velike postoje teškoće kod postavljanja interfejsa 1 pošto je kabl prekratak To se mora rešiti kupovinom novog, većeg kabla

Kod ozbiljnih proizvoda poput ove profesionalne tastatura šok uvek dolazi na kraju — Saga 3 Elite košta 80 funti To je zaista mnogo za dodatnu tastaturu ali verovatno i vredi za onog ko zaista želi da do kraja života radi na svom „spektrumčiću"

B Dj

Prvi krenuo — poslednji stiže

Šuškanja o „spektrumu" od 128K izgleda nisu bez osnove neke softverska firma su već dobile svoja primerke koji su kamuflirani u kutiju starog „spektruma+" sa još



jednom dodatnom pločicom pozadi Novi „spektrum" ima dvojni ROM, tako da će biti kompatibilan i sa starim spektrumom Novosti kod njega ima još odiji bejzik AY 38910 zvučni čip i poodišani displej Još nije sigurno koliko će da košta nova mašina ali je izvesno da će morati da balansira između nova cena QL—a/200 funti/ i previsoka cena „spektruma +" (140 funti)

Jesenja jabuke

Računarska kriza koja upravo hara industrijom računara nije poštedela ni najveće proizvođače Zato se sa velikim zanimanjem prate vesti o tome šta se događa u pojedinim firmama Jedna od najzanimljivijih je, naravno Epl (Apple) koji je zbog svojih pionirskih koraka i zaista senzacionalnih rezultata uvek bio omiljen u rubrikama vesti

Eplu stvari ne idu baš najodlje Od ukupno 6000 zaposlenih otpustio je 1600 ali to je mnogo veća cifra kad se zna da je Epl svoju proizvodnju uglavnom automatizovao pa je i raspon broja zaposlenih relativno mali

Njihovi telefoni

Za sve one koji vole da su u toku kao i za sve one koji su izuzetno ljubopitljivi evo nekoliko zanimljivih brojeva telefona

Acorn Computers (Kembridž) — 0223 210111

Atari (Slou) — 0753 24561

Amstrad (Brentvud) — 0277 228888

Comodore CBM (Northampton) — 0536 205252

Sinclair (Kemberli) — 0276 685311

Ne zaboravite da su sva ova mesta u Velikoj Britaniji i da je za pozivanja Velika Britanija potrebno okrenuti 9944 (i pozivni broj grada bez nule na početku)

Verovatno najveći uzrok krize predstavlja činjenica da polako opada prodaja „epla II" koji je oduvek predstavljao Eplov glavni artikal Na svu sreću po Epl zato polako raste prodaja „mekintoša" (Macintosh) koji je dugo visio nad ponorom propasti Iako tako sve je odlija prodaja laserskog štampača „LaserWrite" i sistema za povezivanja Eplovih računara u lanac AplisTalka-a ipak prodaja ovih proizvoda je daleko ispod proizvodnih kapaciteta

Od silnih glasina koja su kružila o novim Eplovim proizvodima dve su izgleda u rednoj fazi Izgleda da postoje prototipovi „mekintoša" u boji (640×480 piksela) i novog crnobelog „mekintoša" (1000×800 piksela) Epl će izgleda ipak preživeti

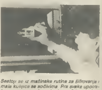
Mali biznis za male pare

Hakeri koji iz noci u noć produkuju bajtova i bajtova budućih dobrih programa rade u velikoj meri Sizifov posao jer je prodaja softvera dobrih programa prilično gorak posao Kada bi bilo moguće formirati male privatne softverska firme za proizvodnju i distribuciju programa plasirati svoj softver na tržište bilo bi znatno manje komplikovano Ali ono što kod nas ne može Englezi su rešili na veoma elegantan način Kod njih je moguće otvoriti svoju sopstvenu softversku kompaniju pa čak i kupiti gotovu već oformljenu softversku kompaniju čije akcije možete da prodajete i tako udete u pravi biznis

Proces kupovine kompanije nije komplikovan — dovoljno je otići kod jedne od agencija za registraciju preduzeća i preuzeti gotova papira za već formiranu firmu Svi troškovi iznose oko 115 funti Na taj način kupujete firmu bez ikakvih birokratskih procedura Ako baš želite da sami osnujete kompaniju i tako uštedite koju funtu, moraćete da potrošite dosta vremena da postignete istu stvar Jedna od pogodnosti takve firme je i ta što u slučaju njenog bankrotstva snosite odgovornost za dugove nastale samo od vašeg dela uloženih para Koliko bi kod nas trajala procedura za otvaranja sopstvene softverske firma, ako bi to uopšte bilo moguće može samo da se zamisli Veoma je verovatno da bi budući vlasnik pre postao penzioner nego biznismen

Antipiratska lupa

Izgleda da se kompanija koja smišljaju zaštitu za programe ne predaju tako lako Najnoviji primer sistema za zaštitu softvera je Lenskok Lenslok je softversko-hardverski sistem koji je mnogo sigurniji od dosadašnjih softverskih zaštita a i jeftiniji je i lakši od dosadašnjih hardverskih brava



Sastoji se iz mašinske rutine za šifrovanja i male kutijice sa sočivima Pre svake upotrebe programa softverski deo zaštite na ekranu prikazuje šifru koja je tako izobličena da se može pročitati samo pomoću pakatića sočiva koji se dobije uz program Pri svakom unošenju programa šifra je drugačija i igra ili program se ne mogu koristiti ako se pre njih šifra ne unese preko tastature

Svaki tako zaštićen program stiže sa svojim setom sočiva koja se ne mogu koristiti ni za jedan drugi program makar bio i od iste firme Naravno, ni ove zaštita nije apsolutno sigurna jer je odgovor u delu programa koji kodira šifru ali je pouzdanija od dosadašnjih rešenja Ako se sočivo ošteti ili izgubi program više ne može da se koristi

Razglednica iz Njujorka ko je najurio džobsa

Dr Radomir A. Mihajlović

Svojim nekorektnim i u svetu biznisa neprihvatljivim ponašanjem Stiven Džobs, jedan od glavnih osnivača proslavljene kompanije Epl nedavno je izazvao toliki gnev upravnog saveta (Board of Directors) kompanije da je bio primoran da podnese ostavku na mestu generalnog direktora

U prelaznom periodu tokom 1980 godina Epl je prerastao iz male radnje za izradu na tržištu dobro prihvaćenih Epl računara u pravu korporaciju zamešnih dimenzija Novcem prikupljenim od akcionara i sa relativno velikim profitima Epl se uvećava do te mere da je racimo samo 1984 godine premašio promet od 1 500 miliona dolara Nasuprot svome drugu iz detinjstva i koosnivaču Epla Stivenu Voznijaku promućurni Stiven Džobs, koji se od samog početka Epl kompanije vrlo malo bavio elektronikom a mnogo više organizacijom i finansijama odlično se uklapa u korporacijsku strukturu „velike Jabuke" Svojom izvanrednom diplomatičnošću organizacionim i govorničkim darom Džobs se održava na samom vrhu Epla sve do pre mesec dana kada je konačno „tikva pukla"

Glavni konstruktor svih prvih verzija „epl računara zaključno sa „eple-II" računarom Stiven Voznijak u razgovoru o svojoj novoj kompaniji CLOUD-9 i problemima koje mu je Džobs ispred Epla pravio ucenjujući zajednička proizvođače repromaterijala sa „ili Epl kao veliki partner ili CLOUD-9? Vozniak pominje Džobsove optužbe da je on navodno šutnuo Epl (reči Džobsa) i da mu zbog toga uzvraća nelojalnom konkurencijom „Oduvek smo bili veliki drugari? Recimo jednom prilikom mi je ponudio da napravim jednu pločicu kasnije poznatu kao Breakout, za Atari arkadu sa igrama Po njegovom predlogu trebalo je da podelimo pola-pola 700 dolara koliko je navodno Atari bio spreman da plati Ja sam napravio pločicu i dobio od Džobsa moju polovinu od 350 dolara i sve do pre par godina dok nisam u jednoj knjizi pročitao nisam znao koliko je moja polovina bila mala Atari je za taj posao platio nekoliko hiljada dolara između ostalog priseća se Vozniak marifetluka Džobsa još iz detinjstva.



Rogovi u vreći Džon Skuli i Stiven Džobs na proslavi Džobsovog rođendana

Džobs je konačno pokazao ko je šutnuo Epl Osnivanjem nezavisne konkurentske firme NEXT inc i vrbovanjem nekoliko ključnih inženjera i menadžera iz Epla prekršio je osnovno pravilo biznis-etikecije u Americi zbog čega je od strane upravnog saveta direktno odgovornim pred akcionerima i investitorima predat okružnom sudu u Santa Klari u Kaliforniji Po optužbi Džobs se iznoseći poslovne tajne i dokumenta uz lažnu lojalnost već duže vremena pripremao da na vrlo unosnom tržištu obrazovnog hardvera i softvera Eplu suprotstavi svoju firmu Iz straha da ne budu i sami optuženi za nemar prema interesima akcionara direktori korporacije se ne usuđuju da zataškaju ovaj slučaj već odlučuju da Džobsu daju otkaz i prijave ga sudu

Odgovor na pitanje „Koliko u optužnici ima istine?" kao i odluku o konačnom rasplatu čitave afere doneće sud u Santa Klari U svakom slučaju nedavne izjave Voznijeka o njegovom odnosu sa Džobsom govore puno u prilog optužbe

U međuvremenu je privredna komora grada Boulder iz Kolorada videla svoju šansu u raskidu Džobsa sa Eplom Džobsu je ponuđeno da uz povoljne uslove prebaci svoju novu firmu u Boulder čime bi se delimično rešio problem nezaposlenosti od preko 7 000 radnika otpuštenih posle bankrotstva nekoliko firmi u oblasti elektronike

Na kraju pitamo se „Ko je, zapravo šutnuo Jabuku — Stiven Vozniak ili Stiven Džobs?", tj pitamo se „Ko je koga tu, zapravo šutnuo?"

SHARP u fazonu IBM-a

Sharp Electronics Corporation je nedavno ponudio tržištu svoj novi portabl mikroračunar PC-7000 Posle uspeha sa svojim minijaturnim mikrećem PC-5000 ovo je drugi značajniji pokušaj Sharpa na rastresitom tržištu računara na kome se džinovi rađaju preko noći ili propadaju za još kraće vreme Sharp se kod proizvodnje PC-7000 uglavnom pouzdao u uspeh IBM-ovog PC računara, kao i u svoje provarene mogućnosti i iskustva na polju mikroelektronike Za razliku od većina mikroračunara sa ekranom od tečnih kristala (na primer proizvodnje Texas Instrumenta Hewlett-packard ili Data General) dodatkom zadnjeg elektro-luminescentnog panela čitljivost kristalnog ekrana Sharpovog računara je drastično uvećana Sharpov ekran takođe obezbeđuje grafiku relativno visoke rezolucije od 640×200 kao i mogućnost prikaza 25 linije standardnog teksta sa 80 znakova po liniji PC-7000 raspolaže sa 320K operativna memorija, (koju je moguće uvećati sve do 704K) i dve disketna jedinice od 5 i 1/4 inča U cilju postizanja kompatibilnosti sa IBM-ovim PC računarom procesor računara je dobro poznati 8088 uz koji je moguće ugraditi opcioni numarički kopar 8087 Ovakvim hardverskim rešenjam Sharp je zaobišao problem koji je mnoge uništio — osim IBM-PC standardnog MS-DOS operacionog sistema PC-7000 ne nudi ništa od sopstvenog softvera, već sa kompatibilnošću sa PC-om u potpunosti oslanja na ogromnu bazu postojećih programa napisanih za IBM-PC.

Od oktobra ove godine Sharpov PC-7000 je u prodaji sa markiranom cenom od nešto malo ispod 2 000 američkih dolara

BBC+128 K

Odmah posle modela B+ Acorn je odlučio da svoj popularni računar dopuni sa dva čipa po 64 K* 4 bita i tako proširi njegovu memoriju do danas sve potrebnijih 128 kilobajta

Memorija može da se organizuje na nekoliko načina najjednostavnije je koristiti računar kao običan BBC B pri čemu ste u startu opremljeni sa 64 K „bočnog" (sideways) RAM-a u koji možete da upisujete programe koji se normalno prodaju u EPROM-ima tekst procesora dodatne programske jezike baze podataka Bazična 32 kilobajte su potpuno slobodne za podatke s obzirom da se video memorija nalazi „u senci" (shadow RAM) Bočni RAM može da se koristi kao bafer za štampač ili, uz dokupljivanje potrebnog softvera RAM disk

Druga je mogućnost da dodatni RAM prekofigurišete i virtuelni adresni prostor &10000 — &1FFFF koji je u celini slobodan za novi bejzik 128 koji dobijete na disku Ova je opcija naročito zgodna za programe koji zahtevaju dosta memorije i malo računanja jer je brzina kompjutera bitno smanjena u odnosu na standardni mod (prem-

Računar	Slobodne za bejzik	Benčmark
BBC Plus/Bejzik 128	64	28 43
Amstrad 464	45.5	14.58
BBC B/B Plus+6502 drugi procesor	44	6,83
Amstrad 6128	41.3	????
Atari 130 XE	40	75.5
Spektrum 48 K/ Spectrum plus	40	58 5
Komodor 128	37 5	????
Komodor 64	37 5	34.0
MSX (npr Sony)	32 (26 sa diskom)	44.3
BBC B Plus	25 5	14.5
Electron	20 5	20.44

da i dalje brža od nekih konkurenata — zbir benčmarke je 29 43)

BBC+128 je kompatibilan sa svim hardverskim dodecima za običan BBC B i sa najvećim delom raspoloživih programa Cena uz ugrađeni disk interfejs iznosi čitavih 500 funti

Za razliku od Amstrada Acorn je na zadovoljavajući način rešio problem onih koji su kupili BBC B plus koji se u budućnosti neće proizvoditi za 30 funti se dokupljuje pločica kojom se BBC B plus u potpunosti pretvara u BBC+128 K nevolja je jedino što tu pločicu treba samostalno zalemiti u kutiju računara

Sto dvadeset osmice trče u krug

Većina proizvođača popularnih osmobitnih računara je nedavno odlučila da dopuni RAM svojih vodećih modela do 128 K i na taj način parira sve jeftinijim šesnaestobitnim mašinama



Ilustracija **Miša Marković**

Ljubitelji računara koji stalno razmišljaju o nabavci novih modela mogu da budu obmanuti tvrdnjom da neki kompjuter ima 128 K memorije uglavnom se očekuje da će ovaj RAM (uz odbitak dvadesetak kilobajta za video memoriju i sistemske promenljiva) biti slobodan za bejzik programe i podatke obradu teksta i slična aplikacije Na žalost, osmobitni procesori mogu da adresiraju najviše 64 K pa je to gornja granica slobodnog RAM-a — ostatak se koristi za video memoriju RAM disk radni prostor CP/M-a

U našoj tabeli je navedeno nekoliko modela 128' i uz svaki od njih količina RAM-a slobodna za bejzik programe i zbir benčmarka 1—8 podatak koji daje predstavu o brzini rada računara Treba dodati da većina kompjutera primenom određenih naredbi postaje kompatibilna sa svojom starijom braćom', uz primetno smanjenje slobodnog RAM-a

Modelima kao što su amstrad 6128 i BBC PLUS 128 osmobitna mašine su po svemu sudeći dostigle vrhunac svoga razvoja Premda će i dalje ostati interesantni za širok krug korisnika osmobitni računari će u budućnosti polako ustupati tržište mnogo moćnijim i ne mnogo skupim šesnaestobitnim naslednicima

Dejan Ristanović

Letenje u učionici

Mikrokompjuterski displej postavljen u učionici igra ulogu realističkog — i jeftinog — trenažera za obuku letače On precizno reaguje na komande pilota podražavajući ponašanje pravog aviona Program predstavlja dragocenu pomoć u obuci iz navigacije i korišćenje instrumenata kao i za vežbanje uzletanje i približavanje zaokreta zaustavljanje i složenih akrobatskih manevara a instruktor (u pozadini) može da ga izmeni dodavanjem tipičnih opasnih situacija turbulencije bočnog vetra mehaničkog kvara otkazivanje instrumenata ili loše vidljivosti izrađen na Univerzitetu u Sautemptonu simulator je popunio prazninu između velikog komercijalnog simulatora i osnovnog programa letenja za kućna kompjutera



Novi uređaj za obuku ima uobičajene kontrolne uređaje i instrumente prikazane na displeju (gornji red sleva nadesno) merač ubrzanja sile teže, indikator brzine vazduha veštački horizont visinomer indikator broja obrtaja motora (donji red) sat, indikator okreta i klizanja kompas indikator vertikalne brzine i indikator napadnog ugla Raspored i kombinacije komandnih uređaja i instrumenata mogu da se menjaju, tako da odgovaraju gotovo svakom avionu ili jedrilici Prikazano obaveštenja na ekranu usaglašava se 16 puta u sekundi, pri čemu se vide i piste za sletanje ili teren „ispod" aviona

Kompjuterski grafički sistem omogućava da se akrobatski manevri uvežbavaju u sporom tempu, kao i da se grafički displej „zamrzne" dok učenik i nastavnik diskutuju o potencijalno opasnoj situaciji i njenom razrešenju Vođa projektantske grupa dr David Olerton (Allerton) koji se vidi za komandama, može da „provuče" avion kroz hangar prikazan na grafičkom displeju »On se nada da će se simulator risije usavršavati za potrebe letačkih škola ili za istraživanja u avijaciji

„Optimalni" monitor

Firma Microvitec' iz Londona nedavno je demonstrirala rad novog monitora CUB 453 specijalno projektovanog i konstruisanog za lična kompjutere CUB 452 pruža izvanrednu rezoluciju i definisanje znakova korišćenjem metoda koji je jeftiniji nego kod dosadašnjih monitora visokih performansi ali znatno bolji nego kod standardnih TV prijemnika Moglo bi se reći da ovaj monitor nudi optimum u pogledu cene i kvaliteta



Kolor sistem niske kompleksnosti koji se koristi u monitoru na osnovu signala za crvenu, zelenu i plavu boju iz kompjutera daje veoma čistu reprodukciju boja Monitor takođe može da prikazuje slike sa video traka i video diskova Ekran je velik 35 5 cm Preko 30 000 škola širom Velike Britanije upotrebljava ovaj monitor za kompjutersku nastavu Monitor je, takođe pogodan za korišćenje kod kuće i u kancelarijama Firma izrađuje modele za rad sa većinom najpopularnijih ličnih računara raznih proizvođača uključujući „Acorn', Apola BBC Researsh Mashines' i Sinclair"

Istočna Meka

Prema pisanju londonskog Tajmse , Sovjetski Savez će uskoro objaviti plan o kompjuterizaciji sovjetskih škola Smatra se da će toj zemlji u sledećih pet godina biti potrebno 1 3 milion malih računara, jer je u planu da se nabavi po 20 računara za 64 0000 srednjih i visokih škola Britanska industrija se tu nada dobrom zalogaju jer je sovjetski mikro-računar agata (epl kompatibilan) zastareo preskup i prekomplicIran za obrazovanu upotrebu

Skoro svaki veći britanski proizvođač je otvorio i prodajno odeljenje za Istočnu Evropu ali za sada prodaja ide polako Ejkorn je nedavno objavio da je uspeo da proda manji broj BBC računara sa mogućnošću povezivanja u mrežu univerziteta u Rigi Mala firma Mamotech prodala je oko 1,000 računara početkom ova godina

Prema mišljenju nekih zapadnih eksperata, Sovjetski Savez radi na toma da svaki učenik u dogledno vreme dobije svoj računar Ako se to dogodi, veliki tehnološki zaostatak za Amerikom mogao bi biti nadoknađen i Sovjetski Savez bi postao svetska kompjuterska sila

A. Zgorelec

Razglednica iz Londona

kriza bez krize

Anđelko Zgorelec

Opšta je ocena da je najgore prošlo — izgleda da industrija mikroračunara polako izlazi iz krize. Kompanije koje su imale prevelike dugove su ili propale ili su promenila vlasnika i sada, poput Ejkorna (Acorn) polako staju na zdravije noge sa dobrim izgledima za trajno ozdravljenje.

Jedino se saga sa Sinklerom i dalje nastavlja poput neke antičke tragedija.

Od velikog spasavanja Sinklera izgleda neće biti ništa. Mnogo dima bez imalo vatre. Izgleda da je Meksvelu više stalo do publiciteta, koji bi dobio učinivši taj korak nego da istinski pomogne britanskoj instituciji ser Klajv Sinkleru (nedavno je Robert Maksvel, izgleda isto tako da dođe na naslovne strane listova, zakupio avion, napunio ga prehrambenim artiklima i lično odneo u gladnu Etiopiju — to je bio Robert Maksvel Aid, bez pomoći Bob Geldofa). Zato, kada su njegovi finansijski eksperti pronašli da Sinkler ima veće dugove od očekivanih, već sklopljeni ugovor, uz sigurno dobru pomoć veštih advokata, bio je raskinut.

Na Maksvelovom cedilu

Sinkler se je našao na cedilu, hrabro je izjavljeno da Maksvel nije ni potreban. Ubrzo je prodata prelepa zgrada Sinklerovog glavnog štaba u Kembridžu (Cambridge) za nekoliko miliona funti, a najvećoj trgovačkoj mreži za prodaju elektronskih produkata Dixon sa blizu 500 radnji u Britaniji utrapljeno je uz povoljnu cenu (bolje da kažemo za bagatelu) računara u vrednosti od 14 miliona funti. Ti milioni funti sada su trenutno izvukli Sinklera iz najveće krize, ali budućnost još uvek nije najsvetlija. Sinkler je, takođe, morao da dozvoli da njegova kompanija za proizvodnju revolucionarnog električnog vozila C5 ode u bankrotstvo jer se ta vozila uopšte nisu prodavala. Sada su u nekim radnjama jeftiniji od dobrog bicikla — svega 99 funti. Sinkler se ponovo busa u prsa izjavljujući da on radi na drugim poboljšanim verzijama električnog vozila. Na zdravlje, želimo mu mnogo sreće! Tu i tamo ipak dolaze i neke bolje vesti iz kompanije čiji je vlasnik prošlogodišnji „čovek godine" (kako slava brzo prolazi). Na važnom španskom tržištu puštena je najnovija verzija Spectruma sa 128K memorije, potpuno kompatibilna sa sadašnjim verzijama. Ova verzija se, za sada, neće prodavati u Britaniji, navodno sve dok Dixon ne rasproda svoje zalihe računara. Takođe, kako saznajemo, Sinkler bi uskoro trebalo da pokaže svoju verziju sve popularnijeg prenosnog celularnog telefona, koji se može direktno priključiti na telefonsku mrežu i to uz cenu od samo 100 funti. Sadašnje cene najjeftinijih modela takvih telefona je oko 1.000 funti. Ukoliko uspe u tome, to bi bio veliki finansijski plus za Sinklera.



Sve mu polazi od ruka. Šef Amstrada Alen Suger pored računara PCW 8256

Kao u starim danima

Vaš dopisnik je jednom napisao da nikako ne bi propustio septembarski sajam „Personal Computer World". Pre svega, to je najveća smotra mikroračunarske industrije u Britaniji. Osim toga, kao osnivač i jedan od organizatora prvih nekoliko sajmova u doba kada je personalna informatika bila u povojima i stvarala prve korake, vezan je za njega i snažnim emotivnim vezama. No, ove godine i nije baš bilo mnogo razloga da se ostaje u Londonu — velika mikroračunarska kriza i najkišovitije londonsko leto ikad zabeleženo oterali su ga na naš Jadran. Ipak, vratili smo se da stignemo bar na poslednji dan sajma, očekujući sumornu i deprimirajuću atmosferu. Još na ulici sreli smo starog poznanika Stiva Englenda, direktora izdavačkog preduzeća koji izdaje nekoliko računarskih časopisa. Na pitanje kako ide sajam, Stiv je odgovorio: „Idi, pa vidi. Kao u starim danima" i zaista bilo je sve prepuno posetilaca — kažu 80.000 — koji su se gužvali oko izloženih proizvoda. A i izlagači su bili hrabri i zakupili mnogo izložbenog prostora. Jedan nedeljni časopis objavio je na naslovnoj strani velikim slovima: „Gde je kriza, kada PCW sajam tuče sve rekorde".

Na sajmu su bili zastupljeni gotovo svi najvažniji proizvođači, ali zvezde su definitivno, bili Atari i Amstrad. Odmah kod ulaza, Atari je imao tako veliki izložbeni prostor da je na prvi pogled izgledalo da je to sajam Atarijevih računara. Jasno da je glavna atrakcija bio najnoviji model 520ST — bilo je izloženo preko 50 modela ovih računara. Čak je i lično Džek Tramiel doleteo iz San Franciska da demonstrira svoje računare. Za ovaj model prikazano je i relativno dosta programa, zatim neke periferne jedinice kao i Winchester tvrdi disk kapaciteta 10 megabajta.

Glavna atrakcija na štandu Amstrada bio je najnoviji računar ove firme PCW8256 kompletan računar sa matričnim štampačem i disk jedinicom i programom za obradu teksta (uz još neke programe) i CP/M operativni sistem za svega 399 funti plus 15 poreza na promet. To je neverovatna cena, jer toliko smo do nedavno plaćali za same štampače. Interesovanje za taj model je bilo ogromno, tvrdi se da je jedna velika mreža radnji želela odmah da otkupi 50.000 komada, a nekoliko velikih kompanija je izrazilo je želju da nabavi ovaj računar za sve svoje osoblje. Jasno da je šef Amstrada u ovom trenutku najsvetlija zvezda na mikroračunarskom nebu. Alen Sugar (Sugar) upravo sija od sreće i on se sigurno pita gde je ta kriza, jer njegova firma je upravo objavila poslovne rezultate koji pokazuju da oni prave veći profit nego ikad dosada.

Na tajnom kanalu

Bolesnik Ejkom, pod italijanskom zaštitom, izložio je verziju BBC-a sa 128K memorije. To je jednostavno 64K BBC sa pridodatom novom štampanom pločom na kojoj se nalazi povećana memorija. Cena ovog hibrida bi trebalo da bude 499 funti. U prodaju će biti puštan u oktobru, dok će stara verzija uskoro prestati da se proizvode. (Na sajmu je i objavljeno da je sklopljen ugovor između Ejkorna, Olivetija i francuskog giganta Thomsona sa ciljem da zajednički rade na novom standardu za računare koji će se koristiti u školama). Ejkorn je pokazao i dosta novih perifernih jedinica, kao i nekoliko robota kontrolisanih računarima ove firme.

Na Sinklerovom štandu bilo je skromnije — samo novi softver i nekoliko novih perifernih jedinica za QL, kao disk jedinica od 3 1/2 inča i štampač za ovaj računar.

Na Komodorovom štandu najviše pažnje je bilo posvećeno C128. „Amiga" nije bila pokazana za javnost, mada su novinari imali priliku da je upoznaju u obližnjem hotelu. Pisac operativnog sistema, mala kompanija Metacomco, imala je jednu „amigu" na svom štandu.

Za razliku od prošle godine, proizvođači MSX računara su izložili svoje proizvode, ali za ovaj, kako bi se to engleski kazalo „stari šešir" nije bilo mnogo interesenata.

Od softvera je, jasno, najviše bilo igara. Izlagale su sve poznatije britanske softverske kompanije — neke su prikazale igre sa spektakularnom grafikom i zvučnim efektima. No, najspektakularnija demonstracija nisu bile na samom sajmu. Da ih vidite trebala vam je magična siva pozivnica za Komodorov hotelski polutajni izložbeni prostor. Uz veoma efektivne igre, ovde je najviše pažnje privlačila simulacija leta za Boeing 737 uz stereo zvučne efekte kao u pilotskoj kabini pravog aviona. Nadamo se da „Računare" ne čitaju u JAT-u, pa da počnu svoje pilote za 737 da školuju isključivo na „Amigi".

Računari u izlogu

živela amiga

Istorija „amige“ je prilično duga i neobična za kompjuterski svet: najpre je Atari osnovao firmu Amiga koje je trebalo da dizajnira grafičke čipova za njihov neuđeni računar Lorraine. Firma Amiga se ubrzo odvojila od Atarija i u tišini nastavila samostalni rad, živeći od kredita koje je neprekidno uzimala od bankara. Predani rad je dao rezultate u vidu projekta za čiju je realizaciju nedostajao jedino novac. U tom momentu se pojavio Commodore, otkupio akcije Amige i tako postao vlasnik istoimenog računara koji ovih dana bez mnogo pompe izlazi na tržište.

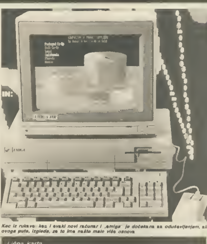
Uz malu krađu ciklusa

Nedostatak pompeznog reklamiranja „amige“ je paradoksalno razlog da verujemo u njegove izvanredne karakteristike. Commodore je u toku ove godine izbacio na tržište nekoliko sopstvenih modela od kojih najjači C 128 nema ni za tranutak smisla porediti sa „amigom“. Commodore se tažeći da proda što više „ato dvedeset osmica“ i tako učini svoj veliki samostalni projekat rentabilnim ne ustručava da pomalo koči „amigu“ verujući da će ona biti jednako privlačna i kroz godinu-dva. Pojave Atarijevog modela 520 ST je međutim naterala Commodora da unekoliko otkrija karte jednostavno pokazujući da je „amiga“ praktično u svakom pogledu superiornija kako modelu 520 ST tako i mnogo poznatijem „mekintošu“.

Amiga se kao i većina modernih računara sastoji od centralne jedinice i tastature. bela lepo dizajnirana kutija veličina 45×33×7 cm sadrži centralni procesor, bitan deo hardvera i slotova za proširenja, među kojima se nalazi i priključak za tastaturu na koju se dalje priključuju džojstik i neizbežni miš. Za one koji su imali priliku da rade sa „mekintošem“ reći ćemo da se na mišu nalaze dva tastera od kojih jedan pomera meni na dole a drugi omogućava izbor neke stavke iz njega. Tastatura je lepo dizajnirana premda se o njenom kvalitetu ne može reći ništa posebno: sasvim je standardna, dovoljna za normalno brzo kucanje ali ne takva da bi vam pred njim zastao dah.

U središtu „amige“ se kao što se može i pretpostaviti nalazi Motorolin mikroprocesor 68000 koji radi na frekvenciji od 8 MHz. Inteligentni čipovi sa romantičnim imanima Daphne (ili Danisa) Agnus i Portia zaduženi su za rad sa sprajtovima, animiranu grafiku i komunikaciju sa periferijom. Zaštitni znak svih ovih operacija je skoro nezamisliva brzina koja je posledica takozvanog „bit blittera“ — posebnog vida direktnog memorijskog pristupa.

Šta je međutim direktan memorijski pristup ili DMA (*Direct Memory Access*)?



Kec iz rukava: kao i svaki novi računar i „amiga“ je dočekana sa oduševljenjem, ali ovoga puta, izgleda, za to ima nešto malo više osnova

Lična karta

Procesor.	Motorola 68000
Clock:	8 MHz
Tastatura:	Profesionalna, odvojena od CPU.
ROM:	192 ili 256 K, u početku 4 K.
RAM:	256 K
Maksimalni RAM	8 M
Spoljna memorija.	Ugrađen disk 3.25 inča, 800 K. Još najviše tri flopi diska (3.25 ili 5.25) Hard disk 10-80 M
Interfejsi:	RS 232, Centronics, video ulaz i izlaz, stereo
Operativni sistem.	Tripos.
Softver	bejzik, Pascal, C, logo (?)
Cena	oko 1000 funti u osnovnoj verziji

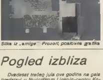
Kod jednostavnih računara kao što je „spektrum“ mikroprocesor ima isključivu privilegiju da pristupa memoriji, ukoliko je, na primer, potrebno da se blok RAM-a prebaci u video memoriju, mikroprocesor će izvršavajući nešto poput LDIR, prebacivati sadržaj bajt po bajt. Zar se može uraditi ikako drukčije? U moderne računare se osim centralnog, ugrađuju i drugi mikroprocesori koji imaju posebne namene. Neki od njih bi u principu mogao da se bavi memorijskim manipulacijama izvršavajući zadatke koja mu je, da bi mogao da se bavi drugim poslovima, poverio glavni procesor

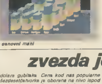
Dok sa „atari 520 ST" gromoglasno najavljuje na sve strana kao mašina iz snova, Commodore je prekrio „amigu" velom tajne koji je počeo da se podiže tek kada su prvi primerci postali spremni za prikazivanje izlazak na svetlo dana izazvao je pravu auforiju među uredništvima stranih časopisa i naveo domaće hakere da pobacaju svoje „galaksije", „spektruma" i „komodore" kroz prozor i ubrzano ubacuju sav džeparac u kasicu Beobanka Uz prikaz „amige" iz pera Dejana Ristanovića, donosimo i viđenje sa lica mesta našeg dopisnika iz SAD dr Radomira A. Mihajlovića

Nevolja nastaje kada su za obavljanja tog drugog posla potrebna nove manipulacije sa memorijom. kako adresnom magistralom može da putuje samo jedna adresa dok se na bseu za podatka može nalaziti samo jedan broj. doći će do konflikte između glavnog procesora i takozvanog DMA kontrolera. Ovaj se problem obično rešava takozvanom kradom ciklusa ako i procesor i DMA kontroler zahtevaju pristup memoriji prednost se daje DMA kontroleru dok mikroprocesor jednostavno čeka da magistrale budu slobodna PARC (Kseroksov Palo Alto Research Centra) u Kaliforniji je predložio bit blitter' kao jednostavno ostvarljivu alternativu kradi ciklusa sasvim pojednostavljeno, glavni procesor i DMA kontroleri obavijaju aktivnosti u alternativnim fazama rada oscilatora Rezultat je zapanjujući dok QL može da promeni 60000 a makintoš 110000 priksela u sekundi amiga' će za isto vrema promeniti čitav milion tačaka! Amigini video kontrolen dakle omogućavaju da mnogo puta u sekundi nacrtate neki lik, obojite ga jednom *od* 4096 boja i da ga onda pomerate po ekranu a da se Motorola 68000 *ava* to vrema bavi nekim drugim proračunima, možda da pnprema nove slika u slobodnim segmentima RAM-a.

Četiri hiljada boja

Skoro neverovatna brojka od 4096 raspoloživih boja je dopunjena izvanrednom ▶







Slika iz „amige" Prozori, poslovna grafika i osnovni meni

Pogled izbliza

zvezda je rođena

Dvadeset trećeg jula ove godina na gala predstavi u Njujorškom Linkoln-centru Komodor je predstavio javnosti svoj novi kontraverzni i po mnogo čemu interesantan mikroračunar amiga Već više od godinu dana računarskim svetom kolaju , poverljivi ' golicljivi' i bombasti detaljčići o vrlo specijalnoj mašini Svečani trenutak je najzad stigao — svetla su indiskretna zvuci barokne muzike uživo ispunjavaju Vivian Beaumont Teatar poznatog Linkolnovog Centra u Njujork Sitiju zavesa je dignuta džambo video projektor se poigrava apstraktnim figurama dok tri operatora u podnožju pozornice sa slušalicama i mikrofonom, nešto ala piloti spejs šatla sa zavrnutim rukavima svečanih odela razmahuju u stilu Artura Rubinštajna po tastaturama dugo očekivanih super računaljki marke amiga" Sve više podseća na multimedija šou nego na vatreno krštenja novog hita kompanija Komodor

Mač nad glavom

Dok se nekoliko TV kamermana provlači između uvaženih gostiju, u vazduhu se oseća treperenje nedavne vesti o finansijskim oavojiama kompanija Commodora International Ltd Komodor navodno očekuje da do kraja četvrtog tromesečja ove fiskalne godine zaokruži cifru od 80 miliona dolara gubitaka Cena kod nas popularne šezdesetčetvorka je oborena na nivo ispod proizvodne cene što je dovelo do projektovanih gubitaka od 50 miliona dolara U drugom tromesečju ove godina izgubljeno je 25 miliona na nepotrebnim operativnim troškovima, što uz izgubljenih 5 miliona na dodatnim periferijskim proizvodima iznosi ni manje ni više već 80 miliona dolara Što se tiče korporacije Komodor svi ti gubici dadu se umanjiti kroz otpisana poreska obaveze Međutim može se samo zamisliti u kakvoj su gužvi i šta sve od AMIGE očekuju glavni kreditori korporacije Komodor Continetal Ilinois Manufacturers Hanover Trust i Barclays Bank Ako bi nedavno predstavljen komodor 128 bio treći amiga bi morala da buda četvrti kec iz rukava — adut koji bi morao da generiše mega dolara kako za kreditora tako i za matičnu kompaniju

Teatar koji prima preko 1700 posetilaca gotovo je napakovan inženjerima i komercijalistima iz Komodora, zainteresovanim distributerima i programerima kao i neizostavnim novinarima Po iznajmljenim napadno svečanim frakovim i beenim' toaletama vrlo je lako prepoznati bilo koga iz familije AMIGA Na tmurnoj pozadini finansijskog pomračenja utegnuti komodorovci izgledaju pomalo smešno

Muzika prestaje i talas tišine prekriva dvoranu Kreatori amige" i tim od četrdesetak inženjera se prestavljaju, a zatim , amiga stupa na scenu Tri velika ekrana u boji prikazuju sva čuda koja može nova mašina Animirane sekvence kolor slike zadovoljavajuća rezolucija, sintetizovana stereo muzika sintetizovani govor i ostale stvari zaustavljaju dah mnogobrojna šarolika publika Šalu na stranu, „amiga" izgleda impresivno

Crtać na „amigi"

,Amiga je lični računar u pravom smislu te reči Predstavljena konfiguracija nije ni kućni računar, ni računar za igre i zabavu, ni poslovni računar već sve to zajedno i još više od toga Utisak je da je marketing ideja poslovnih ljudi iz Komodora bila da se amiga ' ne posvati ni jednom zasebnom segmentu tržišta mikroračunarima, već da se jednom vratom univerzalnosti primene obuhvati što je moguće veći broj potencijalnih kupaca Koncepcijska postavka računara nije originalna kao što to autori AMIGE pokušavaju da provuku ovih dana kroz mnogobrojna natpise u popularnim časopisima o računarima ,Amiga nije gotovo ni u čemu posebno revolucionarna — to se bar za sada ne vidi — osim, u tome što puno obećava i što, po mišljenju mnogih, donosi konačno „pravi" sistem igara onakvog kvaliteta slike i zvuka kakav smo do sada mogli da vidimo samo u specijalnim zabavnim elektronskim parkovima sa skupim arkadnim mašinama Igre os računarima ranijih generacija, kao što je to, recimo, spektrum ' slabom rezulucijom slika i siromašnim repertoarom zvučnih efekata mene su lično uvek više nervirale nego zabavljale Ovo u svakom slučaju, ne važi za „amigu" Sa četiri zvučna kanala, uz ▶

grafičkom rezolucijom 320×200 ili 640×200 U prvom slučaju se iz palete od 4096 boja bira njih 32, a u drugom 16, što znači da će video memorija zauzimati oko 64 kilobajta Ukoliko možete da odvojite više od toga, takozvani *Interlaced mode* će duplirati vertikalnu rezoluciju i povećati horizontalnu, ako vam je potrebno više boja, režim *Hold and Modify* omogućava istovremeno prikazivanje svih 4096 boja, ali u tom slučaju slika mora da bude statična Izvanredne grafičke performanse izazivaju, međutim, i jedan veliki problem: ni za 1000 dolara nije lako pronaći RGB monitor koji bi prikazao kvalitetnu sliku u rezoluciji većoj od 640×256, a u međuvremenu ćete morati da se poslužite najobičnijim televizorom „amiga," osim RGB, kompozitnog video i RGBI (video izlaz na TTL nivou prema IBM-ovim standardima) izlaza ima i RF priključak kao i tajanstveni priključak označen kao video in Uz pomoć ovoga ulaza „amiga" bi trebalo da se takmiči sa daleko složenijim profesionalnim sistemima koji omogućavaju montažu animiranog crteža na običan film

Amiga radi sa dve vrste sprajtova koji su dobili imena „V sprajtovi" (V od virtuelni) i „Blitter objekti" (Boba) Autor ovoga teksta mora da prizna da je vrlo teško shvatiti principe njihovog delovanja i sagledati njihove mogućnosti računar je opremljen sa osam sprajt procesora koji, nezavisno od MC 68000, pokreću praktično neograničeni broj sprajtova željene veličine, brinući se o njihovim kolizijama i izlazima sa ekrana. Virtuelni sprajtovi su pod neposrednom kontrolom hardvera (za njih je zadužen 'Daphne'), što znači da su neobično brzi ali i prilično glupi „Blitter objekti" su sporiji, ali se u svakom od njih mogu naći tačke različitih boje dok je upravljanje daleko jednostavnije — u okviru uputstva za upotrebu „amiga" je dat bejzik program od samo pet linija koji crta King Konga koji, stojeći na vrhu solitera hvata i jede avione koji se motaju oko njega Izgleda da će se RAM od samo 512 K pojaviti kao jedino ograničenje amiginih grafičkih potencijala

Amiga, istina, u osnovnoj verziji ima samo 256 K RAM-a i 192 (potencijalno 256) kilobajta ROM-a RAM proširenja od 256 kilobajta se priključuje sa zadnje strane računara (podseća li vas ovo na samo četiri godine star san vlasnika ZX81 memorijsko proširenje od 16 kilobajta!) Ako vam to nije dovoljno na port za memorijsko proširenje možete da priključite još najviše osam megabajta RAM-a „A šta je sa ostalih osam



„Amiga" i njena anatomija Zahvaljujući čipovima visokog stepena integracije, konstrukcija novog Komodorovog računara je prilično jednostavna

megabajta koja MC 68000 može da adresira?", zapitaće hakeri uvek gladni memorije Ovaj se prostor ne može iskoristiti jer su konstruktori Amiga odlučili da najvišom adresnom linijom kontrolišu periferiju dobijajući time u brzini rada

Računar koji govori

Zvučne mogućnosti amiga ne zaostaju mnogo za grafičkim umesto da iskoristi neki standardni sound čip, Commodore je odlučio da u Portiu ugradi četvorokanalni generator tona sličan čuvenim Fairlight sintetajzerima u memoriju se smešta digitalizovani talasni oblik signala a zatim jedan od stereo zvučnih kanala reprodukuje taj oblik Ukoliko sami ne umete da pripremite digitalizovani talasni oblik, iskoristite običan mikrofon da kažete nešto što će „amiga" izuzetno uspešno ponoviti! Za lakšu sintezu glasa „amiga" je opremljena osnovnim talasnim oblicima muškog i ženskog glasa sa američkim akcentom kojima pristupate direktno iz bejzika. Ukoliko ste ambiciozniji naučićete šta su foneme i sintetisati skoro savršeni ljudski glas koji govori na bilo kom jeziku Ukoliko ste manje ambiciozni, otkucaćete primer iz Uputstva za upotrebu i slušati kako je mogla da zvuči svađa Beki Tačer i Tome Sojera iz čuvenog romana Marka Tvena

U cenu osnovna verzija (oko 1200 dolara) ulazi i jedna Sonijeva disk jedinica pomoću koje se, na disketu od 3.5 inča može upisati čitavih 800 K informacija Jasno je da je kao i kod „atarija 520 ST" kapacitet disketa neusaglašen sa raspoloživim RAM-om — da bi se racionalno koristio jedan kompjutar (pogotovu kompjutar koji omogućava istovremeno izvršavanje više programa) kapacitet spoljne memorije treba da bude 10—30 puta veći od kapaciteta RAM-a što znači da bi „amigi" bio potreban disk od najmanje desetak megabajta — hard disk Na kutiji računara je, zaista ostavljena provizija za hard disk, ali je skoro sigurno da će nju najpre iskoristiti nezavisna firma dok Tecmar poznati proizvođač dodataka za IBM PC uveliko najavljuje dvadeset-megabajtni hard disk sa dva megabajta RAM-a i baterijskim časovnikom za 1000 dolara, Commodore se zadovoljava razmišljanjem o priključenju druge flopi disk jedinice (ovoga puta od 5.25 inča) koja bi zajedno sa potrebnim kablovima, trebalo da košta čitavih 500 dolara pri čemu bi u cenu bio uračunat i IBM PC emulator Ne želeći rat sa IBM-om i verujući da je MC 68000 daleko moćniji procesor od Intelovog 8086 Commodore je, dakle, odlučio da

stereo izlaze sintetizator zvuka i govora sa izlazima za analogni RGB monitor, TV prijemnik kompozitni kolor i monohromni monitor, uz izbor od 4096 različitih boja i rezoluciju od 640×400, „amiga" kao mašine za igre zaslužuje samo komplimente Kompozitorima novokomponovanih, rok i ostalih popularnih melodija još malo pa je odzvonilo Čini se da bi sedamnaestogodišnji heker sa minimumom muzičkog talenta na „amiginom" sintetizatoru mogao da u softveru za dan iskomponuje koliko bi Zamfir ili Arsen mogli u čitavoj karijeri

Uz dodatnu ploču za odmeravanje i kodovanje (digitalizaciju) moguće je „snimiti" zvuk bilo kog muzičkog instrumenta ili reprodukciju gramofonske ploče koja se naknadno može softverski obraditi u vrlo interesantne i originalno izobličene verzije Moguće je da će ne samo muzičari, već i slikari i animatori crtanih filmova videti vrlo korisno oruđe u računaru „amiga"

U okviru demonstracije grafičkih sposobnosti između ostalih reprodukcija visokog kvaliteta, čak i sa alternativnom rezolucijom od 320×200 demonstrirana je animirana sekvenca lopte koja skakuće uz impresivne ehoične stereo zvuke Opšte je poznato da su animirane ozvučene kolor sekvence moguće samo uz ogromne procesorske kapacitete Pored vrlo moćnog Motorolinog 16/32-bitnog mikro-super-procesora 68000, „amigu" ispomažu još tri specijalno za tu priliku projektovana pomoćna procesora iz milošte nazvanih Portia (ulazno/izlazni i akustički procesor), Daphne (video procesor) i Agnes (koprocesor sa kontrolerom brzog memorijskog prilaza) Uz 68000 Agnes je ključna komponenta u obradi animiranih slika Motorolin 68000 je u svetu mikroračunara proslavio „mekintoš" da bi se sada eto, u njega pouzdale i „amiga" pa čak i na velika zvona nedavno objavljeni relativni ekvivalent „amige" „atari" 520ST" Toliko poverenje u 68000 sigurno govori o kvalitetu ovog oprobanog procesora i naši proizvođači mikraća bi trebalo da to imaju u vidu

Romoliki RAM

„Amiga" je kažu, opremljena sa standardnih 256K RAMa, ali je RAM priča nešto malo komplikovanija od toga Naime, Komodor je u početku projekta „amiga" planirao da veći deo operativnog sistema računara, maštovito nazvan „Intuicija" (navodno ga nije potrebno izučavati već je dovoljna i intuicija za njegovu jednostavnu upotrebu) smesti u 192K ROMa Međutim, poučeni iskustvima programera ranijih operativnih sistema, gde važi univerzalni moto da nijedna verzija sistema nije konačna, savršena i bez mane, Komodor i autori „intuicije" iz Velike Britanije se odlučuju da privremeno dok se ne pohvataju sve bubice upisuju sistem sa diska u RAM Šta to znači? To znači da bi ostalo nešto malo više od 64K RAMa korisniku na raspolaganje Rešenje kriznog kapaciteta operativne memorije Komodor je našao u kompletnom posvećivanju 256K RAMa operativnom sistemu Taj segment memorije je nazvan Upisna Kontrolna Memorija (Writeable Control Store), potpuno je zaštićen i ne može se izbrisati čak ni resetovanjem Jedini način da se ovakav ROMoliki RAM izbriše je da se računar isključi iz izvora napajanja Dodatni segment od 256K RAMa je stavljen na raspoloženje korisniku, tako da zvanična konfiguracija „amige" sa 256K u stvari znači 512K od čega je samo polovina direktno upotrebljiva Dalja ekspanzija memorijskog sistema AMIGE je moguća sve do 6,144MB

Operacioni sistem za „amigu" je napisan u jeziku C po specijalnoj narudžbi u Velikoj Britaniji Operativni sistem je multitasking sistem napravljen u dva nivoa tako da intenzivno podržava obradu slike Prvi nivo sistema je takozvani AMIGA DOS dok je drugi nivo pomenuti sistem „intuicija" Uz pomoć DOSa „amiga" može da izvršava više programa simultano u okviru različitih operativnih „prozora" na ekranu Na primer moguće je editovati jedan program dok se istovremeno izvršava program za

pripremi *softverski* emulator koji bi omogućio izvršavanje određene klase programa pisanih za IBM PC na „amigi"!

Čari operativnog sistema

O nastanku i osnovnim karakteristikama triposa, „amiginog" operativnog sistema, govorimo na sledećim stranama „Računara" u okviru teksta Prvi gam za tripos. Zato ćemo ostatak ovog prikaza „amige" posvetiti praktičnim aspektima komunikacije sa kompjuterom i programima koje uz njega dobijate.

Kada pritisnete prekidač koji je smešten na prednju stranu „amigine" kutije na ekranu će se pojaviti slika koja prikazuje disketu koju treba umetnuti u drajv da bi se tripos učitao u RAM. „Amiga" se, dakle, isporučuje kao otvorena mašina u koju po svakom uključivanju, treba unesti operativni sistem, što nikako nije karakteristika kojom se treba preterano oduševiti. Olakšavajuća okolnost je brza komunikacija sa diskom i činjenica da se jednom učitan operativni sistem teško može uništiti. MC 68000 omogućava da se korisniku zabrani upis u pojedine segmente adresne mape što znači da će nehotično „poke" u zonu operativnog sistema izazvati interapt i prijavljivanje greške. Čak ni posle pritiska na ugrađeni taster RESET nećete, kao kod „atarija 520 ST", morati da pretražujete radni sto u nadi da ćete negde pronaći sistemsku disketu.

Verujemo da je isporučivanje operativnog sistema na disketama privremeno rešenje: operativni sistem moćan poput triposa nije ni malo lako debagovati a Metacomco uz to, izjavljuje da bi želeo da čuje komentare korisnika pre nego što finalizuje svoj proizvod i upiše ga u ROM-ove. U „amiginu" kutiju su ugrađena podnožja za ROM-ove u kojima će, možda kroz podinu dana, biti ugrađen konačni operativni sistem i interpretator nekog jezika, na primer bejzika.

Pošto se operativni sistem uspešno učita sa diskete, na ekranu se pojavljuje meni sa imenima programa koji se nalaze na sistemskoj disketi (ukoliko tu disketu zamenite drugom, i meni će se za trenutak promeniti). Negde pri dnu ekrana će se pojaviti i neugledna figurica na kojoj piše „1>". Iskoristite miša da dovedete kurzor u blizinu ovog znaka i pritisnite jedan od tastera na njemu i ova će se pravog programera nedostojna igračka isključiti prebacujući kontrolu tastaturi tako da ćete u budućnosti moći da kucate i ispravljate normalne komande umesto da pokrećete kurzor po ekranu. Time smo objasnili smisao znaka veće (prompta) u oznaci „1>" ali nam je smisao broja „1" i dalje nejasan. On označava da je trenutno aktivan prvi komandni interpreter (CLI) — program koji izvršava naredbe koje izdajemo operativnom sistemu. Obzirom da je „amiga" multiprogramska mašina, za trenutak možemo da otvorimo prozor u kome će se izvršavati drugi komandni interpretator i u kome će se na početku svakog reda pojavljivati znak „2>". Pritiskom na specijalne tastere možemo da se krećemo između raznih komandnih interpretatora i u svakom od njih pokrećemo aplikacije koje su nam potrebne. Svi se prozori, jasno, ne moraju videti — neke možemo u celini da poklopimo drugima, što znači da ćemo moći da startujemo neki program, zaklonimo njegov prozor drugim i u tom drugom prozoru spokojno igramo šah. S vremena na vreme ćemo, naravno, razmeniti prozore i videti dokle je naš program stigao sa poslom.

Veličine prozora i slobodnog RAM-a koji će biti dodeljeni svakom od programa možete ali ne morate sami da birate: iskusni korisnik će najpre podesiti sve parametre a zatim otkucati ime programa koji startuju i tako učiniti obradu racionalnijom dok će početnik otkucati samo RUN (*ime programa*) i prepustiti operativnom sistemu da izabere parametre prema svom nahođenju.

Sledeći tradiciju velikih kompjuterskih sistema, u „amigin" ROM će biti upisana HELP biblioteka: ako ste, na primer zaboravili sintaksu naredbe COPY otkucaćete COPY? i na ekranu će pisati nešto poput COPY FROM TO/A ALL/S QUIET/SIF nedovoljno da saznate sve o ovoj komandi ali sasvim dovoljno da se podsetite šta neki od parametara predstavlja. Comodore je osim ugradnje HELP biblioteke odlučio da snabde korisnika „amige" još jednom pogodnošću koja će se pokazati daleko značajnijom: otvorenim i dobro dokumentovanim operativnim sistemom. To praktično znači da će se počeci potprograma koji obavljaju sve bitne funkcije triposa nalaziti na fiksnim lokacijama memorijske mape i da će svaki od takvih potprograma biti opisan u knjigama koje će se pojaviti možda i pre nego što se računar uopšte nađe u radnjama. Softverskim firmama i ambicioznim korisnicima je ostavljena mogućnost da u okviru BOOT datoteke prestrukturiraju operativni sistem izbacujući njegove delove koji su za neku primenu nepotrebni i ubacujući sopstvena (ili kupljena) proširenja u standardnu biblioteku. Posle ovakve prepravke dobićete novu sistemsku disketu koju ćete ubuduće koristiti posle svakog uključivanja kompjutera.

Imate li 2100 funti?

„Amigin" bejzik je, zapravo, „Personal basic" koji je Metacomco pre dosta vremena napisao za firmu Digital Research. Za izbor ovog bejzika se Commodore-u ne može uputiti previše komplimenata: iako je „Personal basic" dopunjen naredbama koje podržavaju ogromne grafičke, zvučne i multiprogramske mogućnosti „amige", osnova ovoga jezika je danas sasvim zastarela: nema ni pomena od procedura sa prenošenjem parametara, naredba CASE, WHILE petlje i drugih pogodnosti na koje smo se već tako lepo navikli kupci „amige" će se, dakle, podsetiti naredbe GOSUB i

NASTAVAK NA STRANI 40

pretraživanja baze podataka, uz štampanje rezultata prethodno izvršenog trećeg programa. Takve simultane operacije se u literaturi navode kao ili obavljanje-više-zadataka (multi-tasking) ili konkurentna-obrada (concurrent-processing). Pokušana je obrada čak sa 50 simultanih prozora na ekranu. Zanimljivo, kreativnom kombinacijom, mnogobrojnih prozora i „proreza", jednog preko drugog, moguće je ostvariti interesantne trodimenzionalne video efekte.

Za komuniciranje sa računarom, kao softverski interfejs između hardvera i operatera, može se koristiti na relaciji tastatura-računar komandni linijski interfejs nazvan CLI (Command Line Interface) dok se na relaciji tastatura-miš-računar, preko ikona i prozora, može koristiti program „Intuicija". „Intuicija" je napisana po ugledu na operativni sistem poznatog Apple-ovog „mekintoša" i tu, očigledno koncepcijski ničeg naročito novog nema.

Vuk u PC koži

Ne uvek delikatno pitanje softverske podrške „amiga" odgovara vrlo originalno: kompatibilnošću sa IBM-ovim PC računarom, ali ne hardverski, kako je to do sada bilo popularno među klonima PCs već — softverski. Uz disketnu jedinicu za PC od 5 1/4 inča, „amiga" može, za divno čudo, kompletno da imitira PC u softveru koji je najavljen po ceni od pedesetak dolara. Ovakva verzija kompatibilnosti razumljivo moguća je samo sa procesorom tako moćnim kao što je 68000 i iz tih razloga je do sada nismo videli ni kod jednog od poznatih mikroračunara. Brzi i efikasni 68000, drugim rečima, „glumi" Intelov (IBM-ov) 8088 procesor. Sa tačke gledišta korisnika proces emulacije PCa je vrlo jednostavan. Potrebno je učitati sa „amiginog" diska od 3 i 1/2 inča PC emulator program i zatim sa diska od 5 1/4 inča učitati PC-DOS i na ekranu PC standardna poruka

The IBM Personal Computer DOS
Version 2.10 (C)Copyright IBM Corp 1981, 1982, 1983
A>

Posle ovoga bi bilo moguće učitati bilo koji softverski paket napisan za PC.

Softverska emulacija ili imitiranje PCa je kod AMIGE izabrana da bi se zaobišli troškovi popularnih hardverskih rešenja. Naime dosada je u cilju IBM-PC kompatibilnosti uglavnom bilo neophodno ugraditi 8088 ili neki ekvivalentni procesor.

Upotrebom dodatne video ploče (po ceni od približno 200 dolara) moguće je memorisati standardne video slike sa video rekordera, kamere ili drugog računara koje se kasnije mogu programski obrađivati. Kompanija Island Graphics iz Sausalita u Kaliforniji je upotrebom svog softvera i bogatog izbora različitih boja na „amiga" računaru uspela da postigne visoku vernost reprodukcije slike Edgara Degaa čak i sa alternativno nižom rezolucijom.

Diablo kolor inkjet štampač uz „amiga" računar predstavlja u pravom smislu kompjuterizovano oruđe (CAD tool) za izradu slika.

Osnovna konfiguracija AMIGA računara, sa pomenutih 256K RAMa, jednom disketnom jedinicom od 3 i 1/2 inča kapaciteta 880K sa obostranim upisom i bez monitora košta 1 295 dolara. Cena kolor RGB monitora je oko 800 dolara dok je cena dodatne disketne jedinice 3 i 1/2 inče (5 i 1/4 inča takođe) između 350 i 450 dolara. Kada se sve ovo sabere, toliko veličani super-mikro „amiga" sa super-niskom cenom i nije tako jeftin kao što to poslovni ljudi iz Komodora podvlače. Uostalom odakle dolari za čitav spektakl u Linkoln Centru?

Dr Radomir A. Mihajlović

Informacije o AMIGA računaru, kao i o mogućnostima nabavke računara i raspoloživog softvera, mogu se dobiti po ceni od 2 dolara od SOFT ELECTRONICS Inc., 56th st. Brooklyn N.Y. 11219, U.S.A.

pravi programeri ne govore paskal

Hakerski manifest

U starim dobrim danima — „Zlatnoj eri" kompjutera — bilo je lako razlikovati odrasle muškarce od dečaka. Prave ljude od Žderača Pite, kako se to kaže u literaturi. Pravi ljudi su bili oni koji su razumeli kompjutersko programiranje a Žderači Pite nisu. Pravi Programer je govorio „DO 10 I=1 10" i „ABEND" (govorilo se, znate, velikim slovima). Ostatak sveta govorio je ovako: „Meni su kompjuteri suviše komplikovani", ili „Ne mogu da uspostavim odnos sa kompjuterima jer su tako bezlični". U jednom ranijem radu ukazano je da Pravi ljudi nisu ni uspostavljali odnos niti su se plašili „bezličnosti".

Ali, kao i uvek, vremena se menjaju. Danas živimo u svetu u kome male stare dame mogu da imaju kompjuter u svojoj mikrotalasnoj peći, dvanaestogodišnji dečaci mogu da izbace Prave ljude iz vode igrajući „Asteroide" i „Pekmena", a svako može da kupi pa čak i da razume svoj sme-baš-sasvim lični kompjuter. Pravi Programer je u opasnosti da postane iščezla vrsta, da bude zamenjen studentima koji imaju „komodor 64".

Pritisni taster RETURN

Postoji očigledna potreba da se ukaže na razlike između tipičnog visokoškolskog mladog igrača „Pekmena" i Pravog Programera. Ako se ove razlika razjasni, to će onim prvima pružiti nešto čemu mogu da teže — pravi uzor, sliku oca. Takođe će pomoći da se objasni poslodavcima Pravih Programera zašto bi bilo pogrešno da ih zamene dvanaestogodišnjim igračima „Pekmena" (uz znatnu uštedu u plati).

Jezici

Pravi Programer može najlakše da se razlikuje od gomile po programskom jeziku koji koristi. Pravi Programer koristi fortran, a Žderač Pite upotrebljava paskal. Niklaus Virt (Niclaus Wirth), projektant paskala, bio je jednom upitan: „Kako izgovarate svoje ime?" Odgovorio je: „Možete me zvati po imenu, izgovarajući ga kao Virt ili po vrednosti kao Vort (worth — vredan, prim. prev.)". Na osnovu ovog komentara odmah se zna da je Nikolas Virt Žderač pite. Jedini mehanizam prenošenja parametara u programe koje Pravi Programeri prihvataju je POZOVI POTPROGRAM—PRENESI VREDNOST—VRATI SE, kao što je ugrađeno u IBM-370 FORTRAN-G i slične kompajlere. Pravim Programerima nisu potrebne sve apstraktne koncepcije koje je Virt izmislio da bi obavili svoj posao! Oni su savršeno srećni sa starim FORTRAN IV kompajlerom i pivom.



• Pravi Programeri obrađuju ulančane liste na fortranu.

• Pravi Programeri manipulišu sa stringovima na fortranu.

• Pravi Programeri rade poslovne proračune (ako ih uopšte rade) na fortranu.

• Pravi Programeri razvijaju veštačku inteligenciju na fortranu.

Ako nešto ne možete da uradite na fortranu, učinite na asembleru, a ako ne možete na asembleru — onda sve to i nije vredno truda.

Pritisni taster RETURN

Struktuirano programiranje

Akademici u nauci o kompjuterima ušli su u „struktuirano programiranje" tek u poslednjih nekoliko godina. Oni smatraju da se programi mnogo lakše razumeju ako programer koristi neke naročite jezičke konstrukcije i postupke. Oni se, naravno, ne slažu u pogledu vrsta tih konstrukcija i ilustruju svoje ideje primerima od kojih svaki staje na jednu stranu nekog opskurnog časopisa — jasno, jedan primer nije dovoljan da me koga u nešto uveri. Kad sam napustio školu, mislio sam da sam najbolji programer na svetu. Mogao sam da napišem nenadmašan program krstić—kružić koristeći pet različitih kompjuterskih jezika i sastavljam programe od 1 000 redova koji FUNKCIONIŠU (Doista!). Zatim sam prešao u stvarni svet. Moj prvi zadatak u stvarnom svetu bio je da pročitam i shvatim program od 200 000 linija na fortranu, a zatim to ubrzam sa faktorom dva. Svaki Pravi Programer će ti objasniti da ti ni svo struktuirano programiranje na svetu neće pomoći da rešiš takav problem, jer je za to potreban talenat. Neka letimična zapažanja o Pravim Programerima i struktuiranom programiranju.

Pritisni taster RETURN

• Pravi Programeri se ne plaše da koriste GOTO.

• Pravi Programeri mogu da pišu pet strana duge DO petlje a da se ne zbune.

• Pravi Programeri vole aritmetičke IF naredbe, koje program čine zanimljivim.

• Pravi Programeri pišu samomodifikujući kod, naročito ako mogu da prištede 20 nanosekundi u sredini tesne petlje.

• Pravim Programerima nisu potrebni komentari, jer je kôd očigledan.

• Pošto fortran IV nema struktuirani IF, DO ... WHILE, ili CASE, Pravi Programeri ne treba da brinu što ih ne koriste. Osim toga, kada je potrebno, ovo možete da simulirate korišćenjem naredbi poput GO-TO A.

Pritisni taster RETURN

Struktuiranje podataka je takođe dobilo dosta publiciteta. Apstraktni podaci, struktura pointeri liste i stringovi postali su u nekim krugovima popularni. Virt (gorepomenuti Žderač Pite) je doista napisao čitavu jednu knjigu u kojoj je tvrdio da možete da napišete program zasnovan na strukturama podataka, umesto obrnuto. Kao što svi Pravi Programeri znaju, jedina upotrebljiva struktura podataka je matrica. Stringovi, liste, strukture, skupovi — sve su to naročiti slučajevi matrica i mogu lako da se tretiraju na taj način, bez upropašćivanja vašeg programskog jezika svakovrsnim komplikacijama. Najgora stvar sa kitnjastim tipovima podataka je što morate da ih

Neki cinik je programiranje nazvao zamenom za seks, a ovaj tekst, koji tajno kruži među programerima širom sveta, prvim primerkom računarske pornografije. Poput svih škakljivih stvari tekst sadrži zabranjena nemoralna razmišljanja kojima ispravan građanin — to jest struktuirani programer — ne bi trebalo da se bavi.

deklarišete, a pravi programski jezici, kao što svi znamo, imaju automatsko deklarisanje koje se zasniva na prvom slovu imena promenljive.

Operativni sistemi

Koju vrstu operativnog sistema koristi Pravi Programer? CP/M? Bože sačuvaj — na kraju krajeva, CP/M je u osnovi operativni sistem za klinca. Čak i male stare dame i osnovci koriste CP/M.

Pritisni taster RETURN

Uniks je naravno mnogo složeniji (tipičan uniks haker nikada ne može da se seti kako se naredba PRINT zove ove sedmice), ali kada ga se latite i to svim srcem, uniks postaje divna igra. Ljudi ne rade ozbiljne poslova na uniks sistemima! Oni samo razmenjuju kompjutersku poštu preko UUCP mreže i pišu igre avantura.

Ne, Pravi Programeri koriste OSO370. Dobar programer zna šta je IJK3051 greška bez zavirivanja u sopstvani JCL priručnik. Pravi Programer može da piše JCL i bez korišćenja priručnika! Stvarno izvanredan Pravi Programer može da bez heksadecimalnog kalkulatora otkrije bagove zakopane u skladištu od šest megabajta (svojim sam to očima video).

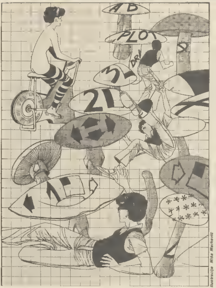
OS je doista izvanredan operativni sistem. Moguće je upropastiti mesec dana posla jednim greškom otkucanim blanko simbolom, pa se u poslu programiranja savetuje oprez. Najbolji pristup sistemu je bušač kartica. Neki tvrde da postoji sistem za rad u raspodeljenom vremenu koji se izvršava na OSO370 ali sam posle pažljivog proučavanja došao do zaključka da su bili u zabludi.

Pritisni taster RETURN

Programske alatke

Kakvu vrstu oruđa koristi Pravi Programer? U teoriji on svoje programe može da izvršava ukucavajući ih na konzoli kompjutera. U ono vreme kada su računari imali konzole, to se u stvari povremeno i radilo. Tipični Pravi Programer je znao ceo „Boot" (program koji učitava operativni sistem sa diska — prim. prev.) napamet u heks-kodu i ukucavao ga kad god bi ga njegov program uništio. U to doba memorija je bila stvarno memorija — nije odlazila u nepovrat kad nestane struje. Danas memorija zaboravi stvari kada to najmanje želite ili ih se seti kada bi već odavno bilo najbolje da su zaboravljene.

Legenda kaže da je Sajmor Kraj (Seymour Cray), pronalazač superkompjutera „kraj" i većine računara firme „Control Data", ukucao prvi operativni sistem na konzoli CDC7600 iz glave čim je prvi put uključen u struju. Sajmor je, ne treba to ni reći, Pravi Programer.



Ilustracije Miša Marković

Pritisni taster RETURN

Jedan od mojih omiljenih Pravih Programera bio je sistem programer firma „Texas Instruments". Jednog dana pozvao ga je jedan korisnik izdataka čiji je sistem krahirao usred presnimavanja jednog važnog rada. Džim (Jim) je bio u stanju da kvar popravi telefonom, tako što je korisniku rekao koje I/O naredbe za disk da ukuca na konzoli da bi korigovao sistemske tablice čiji je haksudakadni sadržaj čitan preko telefona. Naravoučenije: mada Pravi Programer u svoj komplet alatki obično uključuje bušač kartica i linijski štampač, on u slučaju nužde može da se snađe samo sa konzolom i telefonom.

Kod nekih kompanija editovanje teksta se više ne obavlja tako što deset inženjara čeka na red ispred 029 bušače kartica. U stvari zgrada u kojoj sam radio nije pose-

dovala ni jedan bušač kartlosi Pravi Programer je u ovakvoj situaciji morao da obavi posao posredstvom tekst-editora! Većine sistema nudi nekoliko tekst editora i Pravi Programer mora da bude pažljiv i pokupi onaj koji odražava njegov lični stil. Mnogi ljudi veruju da su najbolji tekst editori na svetu napisani u istraživačkom centru „Xerox Palo Alto" za upotrebu na njihovim računarima „alto" i „dorado". Na nevolju nikada nijedan Pravi Programer nije koristio kompjutar čiji se operativni sistem naziva „ćaskanjem" i zasigurno nije se kompjutarom razgovarao pomoću „Miša".

Neki od koncepata iz ovih „Xerox" editora bili su ugrađeni u editore koja su pogonili operativne sisteme sa mnogo razumnijim nazivom — EMACS i VI, da navedamo dva. Problem sa ovim „vidite ono što ćete dobiti" jeste jednako loša koncepcija kod tekst editora kao i kod žena. Ne, Pravi Programer želi „Tražili ste, dobili ste" tekst editor — komplikovan, tajanstven, moćan, nemilostiv, opasan — TECO, da budemo precizni.

Pritisni taster RETURN

Uočeno je da sekvenca TECO komandnih kodova mnogo više podseća na smetnje u vezama nego na razložan tekst. U jednoj od najzabavnijih igara u kojoj se koristi TECO treba da otkucate svoje ime kao direktnu komandu i pokušate da pogodite šta ona radi. Gotovo svaka greška u kucanju dok koristite TECO će po svoj prilici razoriti vaš program ili, što je još gore, ubaciti suptilne i misteriozne bagove u potprogram koji je ranije funkcionisao.

Iz ovog razloga Pravi Programeri nerado prepravljaju fortran program koji gotovo funkcioniše. Po njihovom uverenju lakše je krpiti objektni kod direktno koristeći krasan program nazvan SUPERZAP (ili njegove ekvivalente na mašinama koje nisu IBM). Ovo tako dobro dejstvuje da mnogi izvršni programi na IBM sistemima više i nemaju vezu sa originalnim izvornim fortran kodom. Kad dođe vreme da se prepravi program poput tog, svakom šefu će odmah pasti na um da je najbolje da na taj posao pošalje Pravog Programera — nijedan strukturirani programer Žderač Pite, ne bi znao odakle da počne! To se naziva „siguran posao".

Pritisni taster RETURN

Pravi Programeri NE koriste neke alatke za programiranja:

* Fortran preprocesora kao što su MORTRAN i RATFOR. Kuhinjska majstorija programiranja — odlična za pravljenje pite. Videti prethodni komentar o struktuiranom programiranju.
* Debegera izvornog koda. Pravi programeri umeju da čitaju iz jezgara magnetne memorije!
* Kompajlere koji kontrolišu dimenzije matrica. Oni guše kreativnost, razaraju najveći deo zanimljivih upotreba naredbe EQUIVALENCE i čine nemogućim modifikovanje koda operativnog sistema korišćenjem negativnih indeksa. Najgora od svega, provera dimenzija je bezuspešna!
* Sistema za šifrovanje izvornog koda. Pravi-programer koristi bušene kartice, pa može da čuva svoj program zaključan u ormanu.

Pravi-programer na delu

Gde radi tipični Pravi Programer? Koja vrsta programa je vredna napora tako talentovanog čoveka? Možete biti sigurni da nijednog Pravog Programera nećete zateći u pisanju programa na kobolu ili razvrstavanju pošte za narodni magazin. Pravi Programer hoće zadatke koji doslovce potresaju svet.

* Pravi Programeri rade za Nacionalnu laboratoriju Los Alamos, gde pišu simulaciju atomskih eksplozija na superkompjuteru „Krej 1".
* Pravi Programeri rade za CIA, gde dešifruju sovjetske poruke.
* Zahvaljujući upravo naporima hiljada Pravih Programera koji rade za NASA, američki momci su pre Sovjeta stigli na Mesec i vratili se.
* Pravi Programeri rade za „Boing" gde projektuju operaciona sistema za „krstareće rakete".

Pritisni taster RETURN

Neki od Pravih Programera koji uživaju najveće strahopoštovanje rade za Laboratoriju za mlazni pogon u Kaliforniji. Mnogi od njih znaju napamet čitav operativni sistem na letelicama „Pionir" i „Vojdžer". Kombinovanjem opširnih fortran programa na tlu i malih mašinskih programa na letelici oni su u stanju da izvršavaju neverovatne podvige u navigaciji i improvizaciji — da pogode deset kilometara širok „prozor" na Saturnu posle šest godina boravka u kosmosu ili poprave (odnosno zaobiđu) oštećene senzorske platforme, radio-uređaje i baterije. Jedan Pravi Programer uspeo je navodno da program za analizu spektra ugura u nekoliko stotina bajtova neiskorišćene memorije na letelici „Vojdžer" pomoću njega pretraži prostor i fotografiše novi Jupiterov mesec!

Kao što se zna, mnogi Pravi Programeri na svetu rade za američku vladu — uglavnom za Ministarstvo odbrane. Tako i treba da bude. Odnedavno, međutim, nad horizontom Pravih Programera nadvio se crni oblak. Izgleda da su neki Žderači Pite na visokim mestima u Ministarstvu odbrane odlučili da svi budući odbrambeni programi budu pisani na nekakvom potpuno objedinjenom jeziku nazvanom ADA. Jedno vreme je izgledalo da će ADA imati sudbinu da postane jezik koji će biti u suprotnosti sa svim pravilima Pravog-programiranja — jezik sa strukturama, jezik sa tipovima podataka i obaveznim tačkama i zarezima iza svake naredbe. Ukratko, jezik projektovan da onesposobi kreativnost tipičnog Pravog Programera. Srećom, jezik koji je prihvatilo Ministarstvo odbrane ima dovoljno zanimljivih karakteristika da postane pristupačan — neverovatno je složen, uključujući metode za zamajavanje sa operativnim sistemom i premeštljivom strukturom memorije. Što je najvažnije, Edgar Dikstra ne voli jezik ADA. (Siguran sam da znate da je Dikstra napisao rad „GOTO naredba se smatra štetnom" — kamen međaš u metodologiji strukturiranog programiranja, kome aplaudiraju programeri u paskalu i drugi Žderači Pita.) Osim toga, odlučni Pravi Programer može da piše fortran programe na bilo kom jeziku.

Pravi Programer ponekad nalazi kompromis sa svojim principima i radi na nečemu malo trivijalnijem od razaranja života za koji znamo, pod uslovom da na taj način može dovoljno da zaradi. Postoji nekoliko Pravih Programera koji prave video igre na „atariju", na primer. (Ali ih ne igraju — Pravi Programer uvek zna kako da bije mašinu! U tome nema izazova.) Svako ko radi za „Lukas Film" je Pravi Programer! (Bilo bi glupo odbiti novac od pedeset miliona ljubitelja „Zvezdanih staza".) Udeo Pravih Programera u kompjuterskoj grafici je nešto niži nego što je uobičajeno, pretežno zbog toga što još niko nije otkrio za šta da je koristi.

Budućnost

A šta da se kaže za budućnost? Pravi Programeri su dosta zabrinuti zbog toga što poslednja generacija kompjuterskih programera nije stasala sa istim pogledima na život kao njihovi stariji. Mnogi od njih nisu nikada videli kompjuter sa konzolom. Teško da iko ko danas diplomira može da radi heks-aritmetiku bez kalkulatora. Svršeni studenti su ovih dana neki softverski mekušci — asembleri, disasembleri i dibageri su ih odvojili od programiranja na izvornom nivou, tekst editori im prebrojavaju zagrade, a kompajleri ih štite od svih mogućih grešaka. Što je najgore, neki od navodnih „kompjuterskih naučnika" uspevaju da steknu diplomu a da i ne nauče fortran! Jesmo li mi osuđeni da postanemo industrija uniks hakera i paskal programera?

Pritisni taster RETURN

Iz sopstvenog iskustva mogu jedino da saopštim da je budućnost svugde svetla za Prave Programere. Ni OS0370 ni fortran ne pokazuju znake izumiranja uprkos svim naporima paskal programera širom sveta. Čak su i suptilniji trikovi kao što je dodavanje DO—WHILE i IF—THEN—ELSE—ENDIF struktura fortranu omanuli. Oh, sigurno, neki kompjuterski torbari izašli su sa FORTRAN 77 kompajlerima, ali svaki od njih ima načina da se vrati na FORTRAN 66 jednostavnim izbacivanjem jedne kontrolne kartice.

Čak i uniks za Prave Programera ne mora da bude tako loš kao ranije. Poslednja verzija uniksa ima sva potencijala operativnog sistema vrednog Pravog Programera — dva različita i suptilno nekompatibilna korisnička interfejsa, tajanstven i komplikovan drajver za tastaturu i virtualnu memoriju. Ako zanemarite činjenicu da je „struktuiran", Pravi Programeri cene čak i jezik nazvan „C". Na kraju krajeva, tu ne postoji provera tipova, imena promenljivih su sedam (deset? osam?) znakova duga, a ubačena je i dodatna premija u vidu ukazivača podataka — kao kad bi se najbolji delovi fortrana i asemblera stekli na jednom mestu (da i ne pominjemo razna kreativna upotrebe naredbe DEFINE).

Ne, budućnost uopšte nije loša. U poslednjih nekoliko godine popularna štampa donosila je komentare o novom soju hakera koji napuštaju mesto na Stanfordu i MIT-u i odlaze u stvarni svet. Na osnovu svih činjenica može se reći da duh Pravog Programiranja živi u tim mladim ljudima. Sve dok postoje problematični ciljevi, bizarni bagovi i nerealni rasporedi vremena, postojaće i Pravi Programeri spremni da uskoče i rešavaju Problem, ostavljajući dokumentaciju za kasnije. Neka živi fortran!

NAPOMENA

Ovaj tekst prepisan je sa papira koji je nepotpisan, kružio po Bedfordu. Ima indicija da je autor uniks haker i da mu je korisničko ime ATOIAVEDS MARK. Zna li neko odakle ovo potiče?

Računari u razgovoru sa... prof. dr Radomir A. Mihajlović

o kompjuterima i ljudima

U dosadašnjim razgovorima čuli smo šta o današnjoj situaciji u računarstvu misle naši univerzitetski profesori programiranja. Pokušaćemo sada da saznamo kako ovo stanje vide ljudi koji se istim poslom bave u Americi. Sagovornik nam je naš novi dopisnik iz Njujorka Radomir A. Mihajlović Bili, 35 godina vanredni profesor na njujorškom institutu za tehnologiju. Završio je studije elektronike na beogradskom ETF, magistrirao elektroniku i matematiku, doktorirao i objavio dvadesetak stručnih i naučnih radova iz oblasti elektronike i računarstva.

Mnogi mladi ljudi odlazili su i odlaze u obećanu zemlju Ameriku. Ima i onih koji tamo odlaze zato što žele da se bave naukom za šta kod nas nema dovoljno stručne literature a da i ne pričamo o vrednovanju naučnog rada. Zato su nam odlazila čitava generacija najboljih fizičara a ova godine Katedra za numeričku matematiku i programiranje PMF-a iz Beograda ostala je sa jednim jedinim asistentom jer su se svi ostali sada posvetili razvoju američka nauka. Razgovor sa našim američkim' profesorom Bilijem Mihajlovićem počinjamo pitanjem da li je u SAD otišao sa namerom da se bavi računarima.

— Moram priznati da sam u početku na računare gledao sa dosta omalovažavanja. Ta oblast pre desetak godina kada sam otišao u Ameriku uopšte nije ni bila tako precizno definisana kao kompjuterska nauka. Meni su kao inženjeru usadili mnoga predrasuda i prema matematici i prema računarima i smatrao sam ih samo alatkama koja mogu da mi olakšaju posao. Uostalom, u Jugoslaviji ni danas ne postoje fakulteti za računarske nauke pa nije ni čudo što sam kao i većina korisnika smatrao kompjutersku nauku samo tehnologijom opisanom mnogim receptima uslovljenim realizacijama tih računara. Čovek ili nipodaštava stvari koje ne poznaje ili se boji što zavisi od toga kako mu se prezentiraju.

• **Kažete da ste tek kasnije spoznali da je kompjuterska nauka stvarno nauka?**

— Posle četiri-pet godina bavljenja računarima i predavanja više računarskih predmeta morao sam da saznam više stvari kojima se inače bavi samo mali broj specijalista. Razumevanjem više različitih oblasti i njihovim korelisanjem može se shvatiti njihova celina. Kod nas je još uvek uočljivo veliko omalovažavanje računarskih disciplina. Čuju se i ovakve izjave: „On se bavi operativnim sistemima a to su gluposti, algoritmi koje može da kupi i od proizvođača" ili „On se bavi hardverom a to je majstorski posao. Imaš gotove šeme na osnovu kojih i klinci mogu da sastave kompjuter". Ljudi vrlo površno posmatraju ovu oblast i potcenjuju stvari sa kojima se nisu ni upoznali.

A u Americi ako radiš kao profesor, hteo ne hteo moraš da se upoznaš sa više predmeta. Recimo ja sam prošle školske godine u tri semestra jer mi imamo jesenji, prolećni i letnji semestar predavao ukupno



Računari po meri ljudi prof. dr Radomir A. Mihajlović

deset premeta — možete misliti koji je to širok front.

• **Ali zar to ne ide na štetu kvaliteta predavanja?**

— Razume se da nije svejedno da li će profesor predavati jedan ili deset predmeta i da se na ovaj drugi način ne može prezentirati vrhunska nauka. Ali u Americi se predavanje studentima i ne smatra bavljenjem naukom već pripremanjem obrazovane stručne radne snage za industriju, gladnu industriju koja ih guta kao kit. Predavanja su obična gimnastika, ali vrlo korisna za one koji je upražnjavaju — proširujući i obnavljajući svoja znanja. Ja sam, mada već dugo profesor, svake godine i student.

„Razumevanjem više različitih oblasti i njihovim korelisanjem može se shvatiti njihova celina. Kod nas je još uvek uočljivo veliko omalovažavanje računarskih disciplina."

Posle sagledavanja toliko različitih aspekata o računarima shvatio sam da je kompjuterska nauka za sebe kao fizika ili bilo koja druga priznata nauka i više od toga, studirajući kompjutere saznao sam mnoga stvari o ljudima i njihovom ponašanju. Jer ljudi koriste računare. oni su ih napravili po svojoj meri. Gledajući kroz istoriju računarstva možemo da vidimo da su prvobitni računari bili posvećeni obradi podataka. Druga softverska generacija bavila se obradom informacija, a treća se bavi znanjem. Tu su značajni tzv. ekspert sistemi. Tek u četvrtoj softverskoj generaciji dolazimo do obrade inteligencije. Na ovakav način se razvijalo i ljudsko mišljenje. Kriza u projektima nastupila je zbog nedostatka softverskih oruđa ali i zbog nesazrelosti nauka koje se bave mehanizmom ljudskog mišljenja. Zato slobodno možemo reći da je kompjuterska nauka nauka o ljudima a ne samo o kompjuterima.

• **Danas, i pored velikih sredstava koja se ulažu u projekte veštačkog razuma, nije moguće obrađivati inteligenciju. Kada će i da li će računari uopšte postati mudri?**

— Mislim da će sav taj trud i rad uložen u razvoj računara kada se akumulira dovoljno znanja doneti nove metode koje će se kasnije iskristalisati u neka savršenija softverska oruđa kojima bi bilo moguće obrađivati inteligenciju. Konačno tu nema puno razlike od puta kojim čovek postaje mudar. Počinje od malena sa akumulacijom informacija, dovodi ih u vezu jedne sa drugima i organizuje to znanje. Da bi se nešto organizovalo treba pre svega da imaš šta da organizuješ. Da bi se velika količina znanja organizovala neophodno je prethodno dugotrajno učenje, akumuliranje podataka, korelisanje tih podataka da bi se u jednom trenutku ti podaci organizovali i uopštili u neka univerzalno primenljiva znanja. Tako mudrost može da se smatra jednom višom strukturom saznanja izvan konkretnih manifestacija. Masa konkretnih situacija se uči i kada ih se dovoljno akumulira dovode se u zajedničku vezu, stavljaju se pod zajednički plašt, zajedničku formulu. Čovek koji ima puno tih zajedničkih formula opštepri-

menljivih na pojedine situacije, mudar je čovek. Računar koji će moći da na odgovarajući način sam konstruiše odgovore biće takođe mudar, posedovaće sopstvenu — „veštačku" inteligenciju.

• Znači, po vama, mlad čovek koji nije imao dovoljno vremena da akumulira informacije ne može biti mudar?

— Ne, nikako ne može. On može da bude pametan i kreativan, ali ne može biti mudar. Mudar čovek je onaj koji je u stanju da rasudi kako treba da se postavi u situacijama koje nikad nije upoznao, jer primenjuje mukom stečenu opštu formulu. Kaže: Tako to uvek biva, pa je u ovom slučaju najbolje da se postavim ovako. Samo stari ljudi mogu da budu mudri.

• Zar nije moguće da mudri ljudi prenesu mladima svoja iskustva i tako im omoguće da vrlo rano koriste te, kako vi kažete, opšte formule?

— Ne, ubeđen sam da mladima treba dozvoliti da prođu istim putem kojima su i stariji prošli.

• Meni se čini da je to neracionalno i potpuno neprihvatljivo u današnje vreme. Znam da se i na greškama uči, ali zar nije manje bolno da se uči na tuđim greškama?

— Neracionalno, ali vrlo efikasno. Daću vam dokaz. Dok sam predavao „paskal", mnogo su ga bolje prihvatili studenti koji su prethodno naučili „fortran" ili „bejzik" nego studenti kojima je to bio prvi jezik. Jer oni koji su prethodno naučili neki drugi jezik imali su poštovanje prema finim rešenjima „paskala" koje ne postoje u drugim jezicima i sa uživanjem su ih prihvatali, dok su drugi sve prihvatali zdravo za gotovo bez ikakvog korišćenja sa prethodnim znanjima i bez pravog uviđanja vrednosti rešenja.

• „Kriza u projektima nastupila je zbog nedostatka softverskih oruđa, ali i zbog nerazvijenosti nauke koja se bavi mehanizmom ljudskog mišljenja. Zato slobodno možemo reći da je kompjuterska nauka nauka o ljudima, a ne samo o kompjuterima."

• Međutim, u jednom od prethodnih razgovora za „Računare" profesor Jozo Dujmović je istakao da postoje iskustva na Zapadu koja kazuju da studenti čiji je prvi programski jezik „bejzik" lošije usvajaju moderne programerske tehnike od onih koji ga prethodno nisu učili.

— Ta iskustva mi nisu poznata, ali sam siguran da oni poštuju „paskal" mnogo više od onih koji ne znaju „bejzik" i mnogo će ga bolje koristiti. Recimo, ja sam prvo naučio „fortran", pa „bejzik", pa „paskal", zatim „C" i za svaki sledeći jezik sam imao više poštovanja, jer sam ga dovodio u vezu sa jezicima koje sam prethodno naučio i mogao sam bolje da shvatim prednosti koje donosi.

• Da li je strukturirani način mišljenja koji nudi „paskal" uopšte prirodan za mladog čoveka koji tek počinje da uči programske jezike?

— Nije, jer njegovo iskustvo nije toliko bogato da bi mogao da ga strukturira. Ja sa svojim studentima simuliram, ako je moguće, istoriju razvoja određene oblasti. Zašto su se inženjeri u određenom momentu odlučili za nova rešenja, šta ih je bolelo u vezi sa starim. Studenti taj način vrlo lepo prihvataju, jer žive sa tim inženjerima i njihovim rešenjima kroz istoriju i usvajaju poslednja rešenja koja im se izlažu na kraju semestra sa mnogo većim efektom. Svakako, ne bih preporučio da „paskal" bude prvi programski jezik, jer poseduje strukture s kojima se učenici nikada pre nisu sreli u životu — recimo pojam matrice i višedimenzionih nizova, da ne pričamo dalje. Otuda dete u osnovnoj školi, a uslovi života diktiraju da se početak učenja programiranja mora naći u osnovnoj školi, može da shvati ovaj pojam? Ono ne može da ga shvati jer nema aplikacije za njega. U njegovom iskustvu ne postoje problemi na koje bi primenio višedimenzione nizove i ne može stoga ovakve strukture ni da poštuje ni da koristi. Sasvim je druga situacija kasnije, kada se pojave problemi za čija je rešavanja neophodno upotrebiti matrice.

Po principu izomorfizma mogu se proširiti ova iskustva i na druge probleme u vezi sa računarima u našem društvu.

• Možete li da budete konkretniji?

Hoćemo po svaku cenu da proguramo računare, jer postoji eksplozija interesovanja za njih. Šta delje sa računarima sem igara? Naši ljudi u praksi nemaju dovoljno problema za njih, jer uglavnom malo rade. Pošto malo rada imaju i malo problema i mogu da ih rešavaju i bez računara. A računari nude fantastična rešenja. To su najsavršenija oruđa koja je čovek ikada stvorio. Međutim, sem igara i neke prizemne numerike, većina korisnika kućnih računara uopšte ne vidi čemu oni mogu da služe. Sada se tek kod ponekih oseća potreba za obradom baze podataka i to je već napredak. Ali ako bi se ljudi više radno angažovali, osećali bi i veću potrebu za računarima. Kada takvim ljudima izložiš oruđa za olakšavanje njihovog posla, mislim tu pre svega na programske jezike i softver, oni kažu: Vidi kako ovo jednostavno rešava moje probleme oko kojih sam se do sada tako mučio.

A ako nekom pokazuješ alate koji rešavaju probleme za koje on nikada nije čuo, onda ovu priču on može da prihvati samo kao religiju.

• Koji biste programski jezik vi predložili za ra[illegible]?

Mislim da je najzahvalniji jezik „logo". On je dobar jer grafički nudi informacije a sva dečica vole da crtaju. Treba nuditi jezik koji tretira probleme koji su deci bliski, a to su upravo grafički problemi. Deca, uostalom, najviše i vole kompjuterske igre sa dobrom animacijom.

• „Računar koji će moći da na odgovarajući način sam konstruiše odgovore biće takođe mudar, posedovaće sopstvenu — „veštačku" inteligenciju."

Na drugom nivou treba im ponuditi „bejzik" jer je to interaktivan jezik koji se lako uči. Na ovom uzrastu učenici već imaju i veće probleme, znaju matematiku toliko da im „bejzik" može biti od koristi.

Tek na trećem nivou, negde u srednjoj školi, treba im eksponirati neki strukturalni jezik kao na primer „paskal", koji je izvanredan za akademske krugove, ali se nije proslavio u izgradnji proizvodnog softvera.

Potom već mogu sami da se opredele šta će učiniti dalje u zavisnosti od problema kojima se budu bavili.

To je moje mišljenje vezano za moja znanja. Međutim, opšteprihvaćeno mišljenje u Americi, ali samo na vrhunskim školama, kao što je MIT, jeste da se studentima kao prvi, udarni jezik da „lisp" ili neki „lisp-like", mogli bismo reći — lispoidni jezik koji će im razbiti sve predrasude i pogrešna nasleđa. Tu im odmah demonstriraju komplikovane softverske tehnike, odmah ih šokiraju. Međutim, treba imati u vidu da na vrhunske škole idu samo najkvalitetniji učenici koji mogu da podnesu i ovakve eksperimente.

• A kakvu ulogu u računarskom opismenjavanju igraju kompjuter-amateri?

Slažem se donekle sa onim ko je rekao da su oni ekvivalent radio-amaterima. I sam sam bio jedan od njih. Prvo sam napravio detektor, doduše ne sa cevima već sa tranzistorima. Ništa nisam izmišljao, već sam naprosto zadovoljavao svoju znatiželju, da vidim da li će to što je pravim da proradi. To mi je davalo snage za nove pokušaje koji su rezultirali kroz niz konstrukcija i na kraju doktoratom. Mislim da ovaj hobi ima izvanredno mesto u životu mladih ljudi i da se nikako ne sme potcenjivati. Ne treba od kompjuter-amatera očekivati revolucionarna otkrića, ali ne treba ni isključiti da između njih može da se javi poneki blesak genijalnosti koji dovodi i do novih otkrića. Na stranu što je kompjuteramaterstvo vrlo korisno, jer masira sivu masu, širi vidike i osvetljava nove probleme. Rešenje svakog problema otvara x novih problema i vodi čoveka sve dalje u istraživanju.

• „Hoćemo po svaku cenu da proguramo računare, jer postoji eksplozija interesovanja za njih. Šta delje sa računarima sem igara? Naši ljudi u praksi nemaju dovoljno problema za njih, jer uglavnom malo rade."

Mnogo je korisnije imati mladiće koji imaju probleme i razmišljaju o njima nego mladiće koji ih nemaju i ne razmišljaju ni o čemu. Konačno, oni su [illegible] društva i porodice jer umeju kritički da misle. Kako sam i sam bio radio-amater pa zatim i kompjuter-amater i napravio dva sopstvena računara, mislim da argumentovano mogu da govorim o ovom pitanju. Nisam, doduše, napravio nekakva epohalna otkrića, ali sam sastavio ozbiljne računare, ozbiljnije od većine koji se prodaju u Jugoslaviji, pa čak i od onih koji se koriste u školama. Jedan je bio na nivou „komodora 64", a drugi bolji od njega. Ovaj drugi sam prodao i kupio profesionalni sistem, što mi je omogućilo da se ozbiljnije bavim računarima.

Inače sam u principu protiv centralističkih intervencija, jer mislim da samo haotično kretanje dovodi do kristalisanja pravih vrednosti, smatram da bi, kada se radi o kompjuter-amaterima, bilo korisno da se u svim većim mestima društvenim sredstvima pomažu klubovi kompjuteraša gde bi oni mogli da razmenjuju iskustva, materijal, informacije i literaturu. Ne treba se plašiti amatera, treba ih stimulisati, olakšati im nabavku komponenti, omogućiti im da dođu do potrebne literature. Kompjuter-amaterima treba na svakom koraku pružiti šansu

da se afirmišu jer takvi ljudi su kasnije vrlo korisni u industriji zato što su u stanju da rešavaju konkretne probleme u oblasti elektronike. Dobro je imati široku bazu tehničara elektronike, zato što se one već odavno uvukle u naše domove.

• U ovom trenutku kod nas se pri pomenu kompjuter-amatera odmah pomišlja na pirate. Kako vi, koji radite u zemlji u kojoj su već na snazi i zakoni za zaštitu od piratstva, gledate na ovu pojavu?

— Ja kao humanista, nekakav profesor i intelektualac zastupam stanovište koje ignoriše ekonomski faktor, a stavlja akcenat na stvari koje su od opšte ljudskog značaja i interesa. Mislim da je primarni zakon živeti i preživeti po svaku cenu, a drugi — živeti i preživeti na optimalan način. To znači — živeti sa što manje rada, a što više efekata tog rada i plodova tog minimalnog rada. U tom smislu treba razbiti sve prepreke razvoju čovečanstva, a jedna od osnovnih je prepreka koja sputava razmenu informacija. Sve što stvara šum u razmeni informacija, sve što stvara izvesne prekide i kašnjenje, a to je ekonomski faktor, negativno utiče na razvoj čovečanstva. Informacije treba što brže, što efikasnije i što jeftinije razmenjivati. Softver je novi medijum za prenos informacije i s njim treba postupati kao i sa knjigama. Ja mislim da ne treba nikoga kažnjavati što kopira neku knjigu ako već ne može da je kupi, i u tom smislu ne treba kažnjavati ni za kopiranje softvera.

• Znači, vi mislite da nije krađa kada neko iskopira tuđi vredan i skup program i onda ga krčmi za male pare?

— To je odlična stvar. Nije mi poznato da je neki od pirata koristio tako dobijen novac da se napije u kafani, već ga je uložio u kupovinu novog softvera. To su konstruktivne pare. Mislim da su vrlo napredni klinci koji pored svih barijera i poteškoća omogućavaju brzu i jeftinu cirkulaciju programa.

• Da malo promenim pitanje, jer mi se čini da se nismo razumeli. Kao što ste naporno i dugo, uz mnoga odricanje, radili na svojim stručnim radovima, mogli ste uz isto zalaganje da napravite i neki dobar program. Kako biste se osećali kada bi se plodovi vašeg rada na ovaj način razmenjivali među onima koji nisu u stanju čak ni da procene koliko je tu uloženo znanja i znoja, umesto da za svoj rad sami dobijete odgovarajuću materijalnu satisfakciju?

— Da se to ne može nikome desiti jasno je svima koji poznaju moderne trendove u softver inžinjeringu i marketingu softvera na Zapadu. Uvek ću isuviše da zaradim pre no što drugi počnu da zarađuju na kopiranom softveru. Tu se radi o vremenskom kašnjenju. Udarni profit je vrlo veliki. Čak se možda i od prve prodaje dovoljno zaradi, a ne zaboravite ni ostale.

Za računare nije neophodno znanje matematike, već kritičko znanje, mogao bih reći znanje inkvizitornog tipa, ljubopitljivog tipa. Istina je da matematičari zahvaljujući računarima mogu dobiti mnogo bolje plaćena radna mesta. Međutim, informacije koje obrađuju računari odavno nisu samo kvantitativnog tipa, postoje i druge relacije među strukturama podataka. Po ovom pitanju moj stav se bitno razlikuje od mnogih akademskih mišljenja. Recimo, Dajkstra (Dijkstra) smatra da samo vanserijski ljudi treba da se bave programiranjem, koje je u njegovoj verziji, kreativno stvaranje algoritama za rešavanje problema, a ne pisanje programa. Meni je to u stvari pravo programiranje, algoritmima ne treba da se bave samo eksperti već i obični ljudi, jer će na njihovim greškama i talentovani uočiti ponešto. Svima treba dati šansu da uspeju ili propadnu. Treba bez predrasuda omasoviti kompjutersko obrazovanje. Vrlo je velika opasnost od stroge organizacije ovog procesa, jer postoji mogućnost pogrešnog masovnog usmeravanja, a tada nemamo šanse da dođemo do pravog rešenja. Recimo, Sovjetski Savez, gde se studiozno prilazi problemu obrazovanja, regrutuju se talentovani učenici, šalju se u organizovane škole u kojima ih podučavaju odabrani profesori, nije ništa naročito dao u kratkoj istoriji računarstva. A šta se dešava u Americi? U Americi u kojoj se haotično milioni ljudi bave računarima, i oni talentovani i oni netalentovani, gde svako ima šansu da se oproba na tom polju, pojavljuju se svi glavni pronalasci.

• „Ne treba od kompjuter-amatera očekivati revolucionarna otkrića, ali ne treba ni isključiti da i među njima može da se javi poneki blesak genijalnosti koji dovodi i do novih otkrića."

• Ne bih mogla da se složim sa vama, pronalaske su dali Evropljani, a Amerikancima možemo prepustiti jedino najveći biznis sa računarima. Prvi računar s unutrašnjim programom, prvi poslovni računar, prvi računar sa multiprogramiranjem, prvi računar za široke narodne mase i još mnogo toga engleske su poslastice. Nestranu što su i većinu „američkih" pronalazaka dali Evropljani školovani u svojim zemljama na organizovan način, u organizovanim školama. No, ostavimo se priče o tome ko je koliko do sada doprineo razvoju računara i pređimo na nove puteve njihovog razvoja. Šta vas najviše intrigira u razvoju računara u budućnosti?

— Ono što najviše golica moju radoznalost je simulacija emocije na računaru. Mislim da je u principu moguće na računaru simulirati ne samo inteligenciju nego i emocije. Ako bismo mogli uspostaviti strukturu podataka koji bi sadržavali informacije o nekim fenomenima ali nekompletne, onakve kakvima čovek uglavnom pamti nekakve strukture sa grao okolinom, mislim da bi računari reagovali kao ljudi, tj. emotivno. Ja lično mislim da je koren emocija neznanje, nedostatak informacije i čak nemogućnost da imaš kompletnu informaciju. Kada prvi put sretneš neku osobu, ono što možeš da vidiš je samo minimalni deo informacije koje ti o toj osobi možeš da dobiješ. U nedostatku informacija ti se zaljubiš, a čim prikupiš kompletnu informaciju, ohladiš se. Moje pretpostavke da se emocije mogu zaista ostvariti na računaru podgrava teorija tzv. fazi setova (fuzzy). To su u stvari skupovi sa neodređenim granicama pripadnosti skupu. Kao što su neodređeni pripadnici određenog skupa, tako je neodređena i klasifikacija neke informacije, da li ona pripada ovoj ili onoj kategoriji. Informacije se preplići. Već postoje

• „Informacije treba što brže, što efikasnije i što jeftinije razmenjivati. Softver je novi medijum za prenos informacije i s njim treba postupati kao i sa knjigama."

značajni napori da se ta teorija fazi setova primeni na simuliranje inteligencije. Oni se u modernom softver inženjeringu koriste za projektovanje human-like jezika (jezika sličnog ljudskom) za komunikaciju sa računarima, za projektovanje jezika tipa ljudskog govora. Sve ovo što mnoge moje kolege razbešnjuje mene puno intrigira i ubeđen sam da se primenom nekompletno definisanih struktura podataka može prići simulaciji ljudskog ponašanja na računaru i kreativnom mišljenju.

* * *

Razgovor smo završili sa nadom da, ako ikad i dođe do ostvarenja ovih pretpostavki, to neće biti tako skoro. Čini nam se da su ove stvari suviše daleko od mogućnosti poimanja običnog čoveka i zato se, kao uostalom i većina ljudi, pribojavamo ovakvog razvoja računara. Ali kako je krenulo, nećemo dugo čekati da vidimo da li je Bili Mihajlović bio u pravu.

Razgovor vodila: Navenka Spalević



Periferijska oprema

palice za igru

Većina džojstika radi na jednom od dva osnovna principa. Digitalni džojstik jednostavno zamenjuje tastaturu odnosno njene tipke koje bi se inače koristile za kontrolu smerova. Pomerajte palicu u bilo kom od osam smerova prikazanih na slici 1 i ona će na određen način uspostaviti vezu unutar džojstika, koju će, zatim, interfejs prevesti u kod za odgovarajuću tipku ili kombinaciju tipki sa tastature, odnosno učiniće da centralni procesor „poveruje" da je pritisnuta odgovarajuća tipka. Ako vam ovo objašnjenje daje nadu da možete da igrate svaku igru sa tastature — a neki strastveni igrači tvrde da se najbolji rezultati zbog preciznije kontrole i postižu sa tastature — moramo da vas razočaramo. To je samo delimično tačno jer su u nekim igrama kombinacije koje morate koristiti za kretanje takve da ne bi pomoglo ni kada biste umesto ruku imali pipke kao hobotnica.

Analogni džojstici koriste dva potenciometra — kao kada se na radio-aparatu pojačava ton. Pomeranje gore-dole utiče na jedan potenciometar, a levo-desno drugi. Ali to ne znači da možete pomerati samo u četiri smera! Pomeranje na bilo koju stranu može se razložiti na horizontalnu i vertikalnu komponentu. Vrlo malo komercijalnih igara je projektovano za analogne džojstike — oni se pravenstveno koriste za ozbiljnije aplikacije, na primer za grafiku visoke rezolucije, jer su mnogo precizniji od digitalnih.

Ko mnogo bira . . .

Sve palice za igru su iste? Pogrešno! Kada birate onu koja je za vas najbolja treba da razmislite o sledećim karakteristikama:

— komfor
— brzina reagovanja,
— preciznost,
— izdržljivost i, razume se,
— cena.

Komfor je važan jer ako vas posle pola sata upotrebe zaboli zglob napravili ste loš izbor. Međutim, to je dosta individualna stvar — zavisi od toga koliko su snažne vaše ruke.

Brzina reagovanja je važna u svakoj aplikaciji, a neki džojstici su užasno spori. Bolje su palice koje se same vraćaju u centralni položaj nakon pomeranja u bilo kom smeru.

Preciznost dosta zavisi od brzine reagovanja. Ako je lako osetiti trenutak kontakta kod džojstika, lako je vršiti i minimalna pomeranja koja su potrebna kod mnogih igara. Samocentrirajuće palice čine ovo kretanje „mic po mic" mnogo jednostavnijim.

Izdržljivost je esencijalna. Većina džojstika podnosi dosta grubo rukovanje, ali i pored toga može se desiti da se raspadnu u rukama temperamentnih igrača. Stoga se dobro raspitejte o ovoj karakteristici i za svaki slučaj budite pažljivi.



Mada će pri izboru za većinu biti presudna cena, ipak savetujemo da isprobate više tipova pre donošenja odluke o kupovini, jer svi imamo drugačije ruke i šake. Tako džojstik koji je za vašeg prijatelja idealan može za vas biti sasvim neprikladan. Uz to, treba imati na umu da razne firme prodaju kao svoje iste džojstike, u stvari „atari" ili „spectravideo". Nadamo se da će vam naš pregled dale raspoloživih džojstika i njihovih orijentacionih cena pomoći u izboru vaše palice uživanja.

Raspoložive vrste

Počnimo od industrijskog standarda — palice koju ćete naći u svakoj prodavaonici računarske opreme. **Spectravideo Quickshot I** je proizveden u Americi, a distribuira ga u Engleskoj Vulcan Electronics. To je jednostavan džojstik lepo oblikovane ručke sa tasterom na vrhu i kvadratnom osnovom koja zahvaljujući vakuumskim pričvršćivačima može da se fiksira na ravnoj podlozi. Time se znatno pojednostavljuje rad sa dodatnom kontrolom sa tastature koja je neophodna kod kompleksnijih igara. Ako pričvrstite palicu uz ploču stola oslobađate drugu ruku za komande sa tastature. Ovaj džojstik ima osetljive okidače, dobro reaguje na rukovanje i relativno je jeftin, oko 9.95 funti.

Pored Spectravidea Vulcan distribuira još jedan tip džojstika nemaštovito nazvan **Quickshot II.** I ovaj tip ima pričvršćivače na osnovi i lepo oblikovanu ručku, ali za razliku od prethodnog ima i taster za automatsku paljbu ugrađeno na osnovi. Opcija



Ljubiteljima računara koji su se posvetili igranju video igara nezamislivo je da koriste računar bez džojstika Palice, međutim, nisu „samo za igru". One omogućavaju i jednostavnu kontrolu kursora u velikim menijima, grafičkim programima i programima za obradu teksta

Najpopularniji „spektrum", na žalost, bez interfejsa nije sposoban da radi sa džojstikom, ali bilo koji džojstik sa utikačem D-tipa (standardni Atarijev tip) može normalno da radi pomoću bilo kog interfejsa sa odgovarajućom utičnicom koji je kompatibilan sa „spektrumom" Komodorovci imaju sreću da mogu bez ikakvih posebnih investicija da koriste bilo koji tip zahvaljujući ugrađenim džojstik portovima s desne strane računara

za automatsku paljbu nije možda za igrače „čistunce", jer daje prednosti koje po njihovom mišljenju nisu u okvirima fer igre Kada je pritisnut okidač za automatsku paljbu palica puca onoliko brzo koliko to računar omogućava Tako, ako ste pri uobičajenom uslovima mogli da ispalite četiri hica u sekundi sa automatskom paljbom možete dvanaest No i pored ove prednosti Quickshot II koji košta oko 11 95 funti nije se naročito svideo ljubiteljima igara koji se ređe odlučuju za Quickshot I

Atari džojstik je gotovo istorijski proizvod Mada je veoma jeftin košta oko 7 95 funti zbog lošeg komfora i pored otpornosti na grubu upotrebu, nije favorit

Voitmace Delta 3SC vredi pogledati Ima ravnu pravougaonu osnovu sa tri mala tastera za paljbu od kojih ono u sredini služi za brzu vatru Rukohvat mu je mali prefinjen samocentrirajući je i ima solidnu kontrolu prekidačkog tipa Prvobitno je dizajniran za Dragon računar proizvodi se u Engleskoj, cena mu je 10 funti

Pređimo sada na Wico koji ima više iskustva od bilo kog proizvođača palica Wico palice su značajne zbog odlične konstrukcije i izuzetno visoke pouzdanosti Palica **Bose** koje košta oko 13 funti ima vrlo udobno oblikovanu ručicu sa tasterom za paljbu na vrhu koje brzo reaguje Ručice je od odličnog čelika tako da praktično sve može da podnese Kada bi osnova poput Quickshota imala pričvršćivače za površinu ovaj džojstik bi bio jedan od kandidata za titulu „najbolji džojstik" U Wicovom asortimanu standardni džojstik je **Famous Red Ball, ali s obzirom da košta 23 funte, većini potencijalnih korisnika ostaće nepoznati njegova visoka preciznost i inženjering**

Još skuplji je Three-Way Deluxe džojstik uz koji se dobijaju tri drške koje mogu da se zamenjuju Jedna je oblikovana prema šaci druga je glatka kao palice za bejzbol a treća najviše nalikuje na dršku menjača Mada ima visoke standarde konstrukcije, cena od 25 funti rezerviše je samo za bogate i neodlučne igrače

Iz Suncomove kolekcije koju distribuira Consumer Electronic Ltd dolaze male dobro dizajnirane palice koje su više koncentrisane na kvalitet nego na sjaj **Starfighter** ima malu kvadratnu osnovu sa kratkim držačem zaobljenog kraja Taster za paljbu nalazi se na levoj strani vrha osnove Vrlo je lak za držanje zbog zaobljenih ivica i male težine brz je i precizan, a **košta oko** 13,95 funti

Suncomov **TAC-2** rešava problem levorukih Osnova mu je za nijansu veća i teža nego kod Starfigtera ali ima isti odličan odziv Ima još jedan taster za paljbu na osnovi a košta oko 18,95 funti

Poslednji iz Suncom asortimana, **Joy Sensor**, i nije džojstik — to je džojstik simulator Možda se pitate kako se može simulirati džojstik Joy Sensor je elektronski kontroler osetljiv na dodir I najlaganiji dodir na kružnoj ploči smeštenoj na pravougaonoj osnovi dovoljan je za kreiranje odziva Klizajući prekidač kontroliše da li je pomeranje četvorosmerno ili osmosmerno, što je zgodno za izbor, recimo, između prostornih igara ili lavirint igara kod kojih je izbor smerova ograničen Joy Sensor ima tri tastera za paljbu, pri čemu ono u sredini služi za automatsku vatru, a krajnja su za običnu Pošto ovaj džojstik ne zahteva praktično nikakvu snagu njegov oblik koji deluje nekomforno je potpuno prihvatljiv, mada je potrebno dosta vremena da se čovek na njega navikne Počnite da štedite vaših 29,95 funti odmah jer, pošto nema mehaničkog otpora koji bi trebalo savladati kupovinom Joy Sensora dobijate džojstik kojim se brzo rukuje i koji je praktično neuništiv



Svetski standard: Devetopolni kontaktor „atari"



Digitalni džojstik. Standardna palica za igru omogućuje pokretanje u osam smerova

Upamtite

- Većina interfejsa omogućava da priključite samo jednu palicu Tabela na slici 3 naglašava one koji omogućavaju i ovaj luksuz
- Neki džojstici omogućavaju i brzu paljbu, ali ne može svaki interfejs to i da primi
- Treba proveriti da li interfejs ima sve veze za utičnicu na pozadini „spektruma"
- Budite oprezni ako ste zamenili tastaturu, jer (neki) interfejsi, kao Stonechip Kempston i Cambridge Intelligent Interface, ne mogu da se koriste sa nekima od njih Sve će biti u redu ako je interface i prikopčan van tastature
- U svakom slučaju pre donošenja odluke detaljno proučite karakteristike interfejsa da kasnije ne bi bilo neprijatnih iznenađenja

Interesantno je pomenuti da se za „spektrum" mogu nabaviti i džojstici koji se postavljaju preko tastature i koriste standardnu dršku džojstika za rad sa kursorskim tipkama To su dva vrlo slična i vrlo efikasna džojstika bez elektronike u sebi koji dolaze iz EEc i Grant Design Cena im je 9,95 funti

Kod nas se mogu kupiti džojstici u prodavnicama računarske opreme gde su enormno skupi, ili na oglase Zapad se, ipak više isplati da vam ih neko donese iz inostranstva, jer su tako, i posle plaćanja carine znatno jeftiniji od onih malobrojnih sa konsignacione prodaje

Samo za spektrumovce

Posao interfejsa je da prevede kontakte prekidača iz džojstika u brojeve koji odgovaraju kodovima tipki sa tastature Po načinu rešavanja problema, interfejsi se dele u tri grupe

1 Fiksni interfejsi — omogućavaju da džojstik duplira jedan (fiksni) set tipki ili (kao Kempston) jednostavno pošalje set kodova koji se mogu koristiti u programu za izbor ulaznog porta.

2 „Plug" interfejsi — sa utikačima koji omogućavaju da korisnik izabere koje tipke želi da duplira, i

3 Programabilni interfejsi koji koriste programibilne čipova (PROM-ove)

Izbor razume se, zavisi od toga šta očekujete od svog džojstika Proizvođači fiksnih interfejsa kao Kempston, na primer, pokušavaju da ubede softverske kuće da uključe neophodni kod za njihov interfejs kao opciju u najpopularnijim programima, rešavajući problem diplomatski umesto fleksibilnošću Ako, pak komercijalni program nema opciju za vaš interfejs, možda ima mogućnost za definisanje tipki koje želite da koristite što se svodi na isto, jer tada određujete tipke koje već interfejs može da pokriva. Neki proizvođači daju čak i „konverzione trake" koje omogućavaju računaru da igra igru koja normalno koristi drugačiji set kodova. Međutim, ovo vam ne može pomoći u radu sa softverom koji nije izvorno definisan za rad sa džojsticima

Plug interfejsi su raznovrsni i predstavljaju oličenje jednostavnosti za programiranje Jednostavno uzmete bojom kodirane ili imenovane provodnike (obično sa GORE, DOLE, LEVO, DESNO PALJBA) i povežite ih utičnicom sa odgovarajućim tipkama na tastaturi Ovi interfejsi nisu baš lepi za oko i postoje neke tipke (SYMBOL SHIFT D) koje se ne mogu duplirati, a bile bi korisna u programima za obradu teksta Takođe, morate da prespajate veze kada koristiti program sa drugačijim setom komandnih tipki.

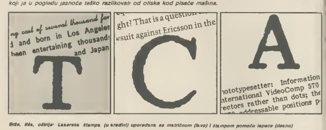
Programabilni interfejsi su veoma raznovrsni i jednostavni za ugradnju, ali, naravno, fleksibilnost i usavršena elektronika imaju svoju cenu Pošto su skuplji od drugih tipova, na vama je da donesete odluku da li vam njihove opcije trebaju

Nevenka Spalević i Vladimir Ilijevski

Periferijska oprema

laserska veza

Sva donedavno dva najviše korišćena tipa štampača bila su matrični i printer sa lepezom. Prvi je brz i bučan i uglavnom daje štampu lošijeg kvaliteta. Drugi nudi štampu koja se može uporediti sa ovom kod pisaće mašine, ali je spor, bučan i nesposoban da se pozabavi grafikom. Pred nama je sada novi izbor - laserski printer. Laserski štampači su mnogo brži od najbržih matričnih printera, izgledaju tihi i daju otis koji je u pogledu jasnoća teško razlikovati od otiska kod pisaće mašina.



Brže, tiše, oštrije: Laserska štampa (u sredini) upoređena sa matričnom (levo) i štampom pomoću lepeze (desno)

Tehnika nije nova: laserski štampači firmi „Xerox" i IBM nalaze se u prodaji već šest godina. Razlika između njih i novog printera je u tome što ovaj ima i prihvatljivu cenu. Raniji laserski printeri su po pravilu koštali najmanje 20 000 dolara. Zahvaljujući tehnološkom prodoru firme „Canon", vlasnici kućnih računara mogu sada da kupe laserski štampač za 3 500 do 10 000 dolara. To može da izgleda mnogo, ali treba imati na umu da su štampači sa lepezom toliko koštali pre dva ili tri godine, danas najjeftiniji iznosi nekoliko stotina dolara.

„Cena laserskih štampača će zasigurno pasti u skoroj budućnosti", kaže Robert Grandžer (Granger), direktor proizvodnog marketinga za laserski printer firme „Hewlett-Packard", poslednjeg od skupih uređaja ove vrste. Odlične verzije prave i „Quality Micro Systems", „Imagen" i nekoliko drugih kompanija.

Laserski štampač unekoliko radi kao aparat za kopiranje. Kod njega se na površini obrtnog valjka koma je dato pozitivno naelektrisanje formira odgovarajući lik. Valjak se zatim presvlači suvim praškastim „mastilom" i lik se prenosi na papir.

Kopir-aparat formira celu sliku sa originala odjednom, dok laserski printer jedno vreme pravi jednu liniju, na osnovu digitalnih podataka koji predstavljaju slovne i grafičke simbole predviđene za štampanje. Kada se skup podataka za čitavu stranu prikupi u baferskoj memoriji, infracrveni laser stupa u dejstvo.

Vođen jednim elektronskim uređajem za kontrolu, laser se pali i gasi hiljadama puta u sekundi, dok njegov zrak poigrava preko valjka. Svaki put kada precizno fokusirani zrak pogodi površinu osetljivu na svetlost, pozitivni naboj se lokalno neutrališe.

Rezultat je šablon neneutralizovanih tačaka, koje se prepliću i formiraju slova i cifre u red, mora pozitivnih naboja. „Mastilo" (zapravo prah od crne plastike) takođe je pozitivno naelektrisano. Kada se nanese preko valjka, istovetni pozitivni naboji se međusobno odbijaju, i prah se zadržava jedino uz nenaelektrisane tačke. Valjak se okreće preko papira, prah se prenosi, a grejač stapa plastiku na hartiji.

Laserski zrak treba da se kreće toliko brzo — čitavu stranu pretvara u sliku za oko sedam sekundi — da bi bilo nepraktično sam laser pomerati tamo-amo. Umesto toga, fiksirani zrak se reflektuje sa površina ogledala sa šest ploha koje se vrti sa nešto preko 93 obrtaja u sekundi. Dok jedna strana ogledala promiče, ona odražava zrak prema valjku sa neprekidno promenljivim uglom. Valjak se takođe okreće, spuštajući stranu, dok se zrak kreće preko nje.

Pošto se kod njega slika formira od tačaka, laserski printer može da se klasifikuje kao matrični štampač. Ali tačke su toliko male — na svaki kvadratni centimetar staje ih preko 30 000 — da ih je gotovo nemoguće videti, čak i ako se koristi lupa. Konvencionalni matrični štampači, koji tačke formiraju udaranjem vrhovima igličа u vrpcu, ne mogu to da urade toliko dobro.

Za raniju skupoću laserskih printera kriva su dva razloga: prvi — helijumski laser standardnih osobina bio je skup, drugo — metalni valjak za štampanje mogao je lako da se ošteti, a zamena je bila skupa.

„Canon" je problem rešio upotrebom relativno jeftinog, dugotrajnog i kompaktnog lasera i kopiranjem svog vlastitog kopir-aparata. Plastični valjak u laserskom štampaču je već bio na raspolaganju kao i mehanizam koji razdeljuje plastično mastilo. Oba se nalaze u jednoj jedinstvenoj, lako zamenljivoj kaseti. Kada se valjak i mastilo istroše — obično posle štampanja 3 000 strana — kaseta se jednostavno zameni, po ceni od 99 dolara.

Da li i druge poslovne kuće pripremaju svoje vlastite laserske štampače? IBM, kao i uvek, odbija da to prokomentariše. Tom Hiber (Hieber) iz „Xeroxa" jedino kaže: „Razložno je pretpostaviti da se osvrćemo naokolo kako bismo proširili naše proizvodne kapacitete".

Postoji i nekoliko manjkavosti laserskog štampanja: ne može da se radi sa višeslojnim formama jer nema udaranja, ne može da se štampa u boji, a printeri su toliko brzi da je problem da kompjuter drži korak sa njima. „Moraćemo da se ponesemo sa poslednja dva problema", kaže Grandžer, „ali nema sumnje da ćemo ih rešiti".



Priprema Dejan Ristanović

Kriptografska zagonetka

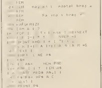
Računari se između ostalog mogu koristiti i kao izuzetno moćne mašine za šifrovanje poruka ali i kao alatke za probijanje raznih postojećih šifara. Zato smo odlučili da vam u sklopu našeg drugog nagradnog zadatka, predstavimo šifru koju je koristio Julije Cezar za prenošenje naredbi svojim legijama.

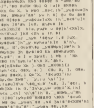
Cezar je svakom svom generalu davao po jednu tablicu slovnih zamena u kojoj bi, da se tada koristila naša abeceda pisalo da se slovo A zamenjuje slovom M slovo b slovom Q tačka slovom J i slično — svakom bi slovu ili znaku bio jednoznačno dodeljen neki drugi element azbuke. Slovo po slovo čitava se poruka mogla šifrovati a onda primenom iste tablice i dešifrovati.

Mi živimo nekoliko stotina godina posle Cezara pa je red da ovaj posao malo osavremenimo — napisali smo program sa slike i koji najpre u elemente niza A smešta zamene za svaki ASCII znak (redosled ovih zamena je jasno zavistan od konstante koju napišimo iza RANDOMIZE i koju smo zamenili trima tačkama), a zatim sa tastature uzima poruke šifruje ih slovo po slovo, a onda štampa rezultate. Zatim smo primenili program i šifrovali nešto izmenjen početak poslednjeg poglavlja jednog po našoj oceni izvanrednog naučno-fantastičnog romana i priložili rezultat. Tekst je kucan uz korišćenje velikih i malih slova naših č ć, ž i š normalne interpunkcije (računajući zareze zagrade i slične znakove) ali bez novih redova ili bilo kojih drugih kontrolnih znakova.

Vaš zadatak nije jednostavan — dešifrovani tekst dopunite kuponom koji objavljujemo i sve to pošaljite na adresu „Galaksija (nagradni zadatak 2) Bulevar vojvode Mišića 17 Beograd pre 15 decembra 1985. Među tačnim odgovorima će biti izvučene tri novčane nagrade 10000, 5000 i 2000 novih dinara.

Kako da pridete rešavanju problema? Najbolje je da prebrojite (pomoću kompjutera naravno) koliko se puta u tekstu pojavljuje koje slovo pa da za najčešće korišćeno pretpostavite da je malo a





```
80 DIM DANI$(6
90 FOR I=0 TO 6:READ DANI$(I):NEXT I
100 INPUT "Unesi broj Računara";N
110 IF N=0 THEN END
120 IF N 3 THEN PRINT "Tada su Računari još izlazili neredovno":GOTO100
130 D=10
140 M=(N-3) MOD 6*2
150 IF M=0 THEN M=12
160 G=(N+2) DIV 6+1984
170 PRINT "Računari ";N;" izlaze ";
180 PRINT ;D;".";M;".";G;" godine -
190 GP=G+(M 2 05)/12
200 JDN=INT(INT(INT 367*GP) INT(GP)-0 75*INT(GP)+D)-0 75*INT(GP/100))+1721115
210 PRINT "u ";DANI$(JDN+1) MOD 7);
220 GOTO 100
230 DATA nedelju,ponedeljak,utorak
240 DATA sredu,četvrtak,petak,subotu
```

Ime i prezime	
Godina rođenja	
Adresa	
Mesto	
Vreme rešavanja zadatka (časova)	

Nagradni zadatak
Mnogo tačnih odgovora

Naš prvi nagradni zadatak očito nije zadao previše problema čitaocima „Računara" — u predviđenom roku je pristiglo preko 90 odgovora, većinom tačnih. Naše rešavače su mučile jedino dileme nevezane za samim problemom.

Malo podsećanja neće biti na odmet prvi nagradni zadatak je tražio odgovor na pitanje u koji dan izlaze „Računari" broj 101". U zadatku je jasno napisano da „Računari" izlaze svakog drugog meseca ali se po Marfijevom zakonu dogodilo da zadatak bude objavljen baš u prošlim „Računarima" kada smo najavili prelazak na mesečni kalendar izlaženja. Svi su rešavači naravno poštovali uslove zadatka ali telefonskih pitanja nije bilo malo. Osim toga broj 101 mnogo liči na binarni broj pa su se mnogi pitali da se ne radi o zamci možda smo tražili datum izlaska „Računara 5"? (kada smo već kod toga da pomenemo jednog čitaoca koji je zaslužio nagradu za glupost meseca došao je do zaključka da je broj 101 napisan u binarnom sistemu a onda ga konvertovao u dekadni broj 9???)

Druga nevolja sa zadatkom je što smo najavili rezultate za „Računare 9" koji su 1 novembra po isteku predviđenog roka bili praktično zaključeni imali smo vremena jedino da otvorimo pisma, letimično pogledamo programe i grupišemo rešenja u tačna i pogrešna. Među tačnim rešenjima koja su zajedno sa originalnim kuponima, prispigli u Redakciju pre 1 11 85 izvukli smo novčane nagrade od 5000 i 3000 dinara koje su dodeljene ***Slobodanu Mazeku*** *iz* ***Mlađenovca*** *i* ***Goranu Liliću*** *iz* ***Zaječara.***

Prva nagrada (10 000 dinara) je bez mnogo razmišljanja dodeljena(Miloradu Zatezalu iz Maribora (ako nas pamćenje dobro služi aktivnom članu našeg bivšeg Kluba korisnika džepnih programabilnih računara) Drug Zatezalo je napisao divan program za Amstrad CPC464 (oko 130 linija) koji rešava problem uz pesmu i grafiku. Program je bio od veoma retkih koji predviđaju da „Računari" ne mogu da izađu ni u subotu ni u nedelju.

Na slici je dato jednostavno rešenje nagradnog zadatka testirano na računaru BBC B. Posle primanja podataka linije 130—160 izračunavaju datum izlaska zadatih „Računara" dok linije 190—210 koristeći zanimljivu formulu koju su razvili članovi američkog PPC kluba izračunavaju traženi dan u nedelji. Izvršavanjem programa saznajemo da „Računari 101" izlaze u utorak 10. 4 2001 godine.

Ovaj napis završavamo sa nekoliko preporuka za rešavače naših sledećih problema uvek rešavajte zadatak onako kako je postavljen bez obzira na napomene u nekim drugim tekstovima. Uz rešenja obavezno pošaljite kupon (izvlačenje je pošteno jedino kada se u bubnju nalaze samo jednaki kuponi) pričvršćen na nekom vidljivom mestu. Pismo sa rešenjem ne treba da sadrži druge komentare, pitanja i zahteve jer je vrlo verovatno da ovi neće biti pročitani. Na kraju ne šaljite kasetu osim ako se radi o nekom veoma specijalnom rešenju, ukoliko se odlučite da pošaljete kasetu i očekujete da je dobijete natrag, obavezno priložite ne sebe adresiran čvrst koverat sa dovoljno marki.

Dejan Ristanović

koma zbog komala

JEZIK IZ (PROGRAMERSKIH) SNOVA

Komal je najjednostavnije predstaviti dobrome poznavaocu postojećih kompjuterskih jezika: zamislite sve dobre osobine fortrana 77 (osim kompleksnih promenljivih), ujedinjene sa svima što vredi u paskalu, a onda sve to svedite na jednostavnost bejzika i — imaćete predstavu o mogućnostima komala!

Komal je jezik koji se interpretira, što znači da kucate program a zatim, komandom RUN, započinjete njegovo izvršavanje. Ubrzano se pripremaju komal kompajleri pomoću kojih ćete kada jednom završite razvoj programa i potpuno ga testirate, prevesti svoje ramek-delo na mašinski jezik i tako ga višestruko ubrzati. Obzirom da se jezik interpretira, za kucanje programa se koristi editor sasvim sličan bejzikovom: linije nose brojeve i uređuju se po njihovom rastućem redosledu.

Inteligentni editor

Sintaksni analizator se aktivira posle unošenja bilo kog programskog reda i prijavljuje sve greške. linije će ući u program samo ako je potpuno ispravna. Linija, međutim ne ulazi u program u obliku u kom ste je otkucali. napišite na primer, 10 FOR PETLJA=1 TO N i pritisnite RETURN, a zatim izlistajte program

10 FOR petlja :=1 TO n DO

Komal je, pre svega napisao naredbe (FOR, TO, DO) velikim a imena promenljivih (petlja n) malim slovima. Dodao je, osim toga dve tačke ispred jednako naglašavajući da se radi o dodeljivanju a ne o jednakosti (šta bejzik radi kada napišete B=4 C=6 A=B=C PRINT A? Zašto?) i dopisao službenu reč DO koja je pozajmljena iz paskala. Mogli ste, naravno, da otkucate naredbu onako kako je kompjuter želi, ali nema nikakve potrebe da se nepotrebno mučite!

Ako pogledate primere programa koje dajemo uz ovaj tekst, videćete da su komal programi uredno „nazubljeni" naredbe koje čine petlje i potprograme su uvučene za po dva blanko simbola petlje u petljama za još dva i tako dalje. O ovom nazubljivanju ne razmišlja programer naredbe kucate kako želite a komal interpreter ih pomera pri listanju pri čemu vodeći blankovi ne zauzimaju nikakav memorijski prostor! Da bi ova kozmetika mogla da se ostvari, autori komala su zabranili smeštanje većeg broja naredbi u jednu liniju, što je jedna od svega dve zamerke koje uz svu moguću kritičnost možemo da uputimo komalu

Promenljive i aritmetika

Komal razlikuje celobrojne, racionalne, logičke i znakovne promenljive, a standar-



dom je predviđeno i postojanje promenljivih u dvostrukoj tačnosti koje se u implementacijama obično zanemaruju. Imena promenljivih se sastoje od proizvoljnog broja slova (ako koristite velika slova, računar će ih pretvoriti u mala) pri čemu imena celobrojnih promenljivih treba da se završavaju znakom # (taraba ") a imena stringova znakom $ (dolar) Mogu se definisati nizovi i matrice do osam dimenzija — daleko više nego što će dopustiti memorija vašeg računara pa makar se on zvao i „amiga"

Rad sa celobrojnim i racionalnim promenljivima je sasvim standardan, kao u bejziku. Vredi jedino pomenuti da se umesto a :=a+x može napisati a :+x, a umesto a :=a−x, naravno, a :−x. Logičkim promenljivima se dodeljuju vrednosti logičkih izraza u kojima smeju da se koriste operacije AND, OR, EOR (ekskluzivne disjunkcije) i NOT

String promenljiva ako drukčije ne naglasite pamti najviše 40 slova. Dimenzionisanjem (npr DIM a$ OF 120) možete da ih produžite ili (DIM a$ OF 5) skratite. Vlasnici Sinklerovih računara će biti počastvovani (ako tu počast ne pomuti tvrdnja tvoraca komala da su koncipirajući rad sa stringovima imali u vidu fortran 77) kada pročitaju da komal ne poseduje funkcije LEFT$ RIGHT$ i MID$, umesto njih, delovi stringa se referenciraju navođenjem koordinata slova. Naredba a$:=b$ (3:5) će u a$ smestiti treće, četvrto i peto slovo stringa b$. Da bi stvar bila lepša a$:=b$ (3: −3) će izdvojiti podstring niza b$, počevši od trećeg slova pa do trećeg slova sa kraja! Za izdvajanje pojedinih slova se može napisati a$:=b$ (3:3) ili i a$:=b$ (3) pri čemu je blanko između dolara i otvorene zagrade bitan, da smo ga izostavili računar bi izdvojio treći element niza B$ a ne treće slovo promenljive b$

Dimenzionisanje nizova je pozajmljeno iz bejzika pa će posle DIM a$(100) postojati a$(0) a$(1), ... a$(99) i a$(100). Mogli smo, međutim da napišemo i DIM a$(−20: 30), posle čega bi postojali a$(−20), a$(−19) ... a$(0), ... a$(29) i a$(30). Za izdvajanje podstringova kod elemenata niza se koriste na prvi pogled čudne ali vrlo efikasne konstrukcije a$(3)(7:10) := b$(4)(5:8). Kako je moguće zameniti četiri slova (sedmo do desetog) sa dva (peto i šesto)? Komal, za razliku od mnogih bejzika, omogućava da se string pri zameni svojih delova produžava ili skraćuje!

Kontrolne strukture

Izvanredno rešen rad sa promenljivima i stringovima je samo jedna od komponenata vrednosti komala — druga i možda daleko važnija komponenta su kontrolne strukture. Komal umesto bejzikovih IF ... THEN i FOR ... NEXT struktura ima IF ... THEN ... ELIF ... ELSE ... END IF, CASE ... WHEN ... OTERWISE, FOR ... NEXT, REPEAT ... UNTIL i WHILE ... END WHILE. Delovanje ovih kontrolnih struktura je prikazano u programu sa slike 1

Najjednostavnije je razumeti naredbu IF: ako je uslov iza nje ispunjen izvršava se naredba ili grupa naredbi iza THEN a zatim se prelazi na naredbe iza END IF. Ukoliko uslov nije ispunjen izvršava se sekvenca iza ELIF (od ELSE IF), pri čemu se tu može naći novo THEN ... ELIF. Ukoliko ni jedan od uslova nije ispunjen izvršava se (ako

[illegible]

postoji) naredba iza poslednjeg ELSE, a zatim se prelazi na ostatak programa iza END IF. Ukoliko vam je teško da se naviknete na ovu komplikaciju možete da koristite skraćeni oblik naredbe IF pa da kao u bejziku napišete IF a> b PRINT "Veće je" — računar će sam umetnuti potrebno THEN

Bejzik, svakako najpopularniji među svim kompjuterskim jezicima, poslednjih godina preživljava teške dane — anatemisan je kao nestandardizovan, nestruktuiran, nepogodan za učenje, spor… Nije mu, međutim, lako lomiti kičmu — ni jedan računar ili operativni sistem se danas ne može zamisliti bez bejzik interpretera. Autori modernih bejzika se svojski trude da pojačaju svoje proizvode dodavanjem novih kontrolnih struktura, proširivanjem seta naredbi i konstrukcijom brzih interpretera i prevodioca. Tako nastaju superbejzici koji su, međutim, međusobno sve nekompatibilniji. Na sreću, još prilično davne 1974. Danci Benedikt Loefstedt Loefstedt i Bord Christensen zamislili su komal (Comal) kao univerzalno i standardizovano proširenja bejzika još nedovoljno definisanih karakteristika. Pojava BBC-javog bejzika je ubrzala standardizaciju komala, pa je maja 1982. objavljen takozvani komal 80 standard, da bi se tokom prošle godine pojavile verzije komala za bbc računare, electron i komodor 64

U nekim je slučajevima daleko komfornije koristiti strukturu CASE njome testiramo nekoliko vrednosti nekog izraza i, ako pronađemo onu koju smo tražili, izvršavamo odgovarajuću grupu naredbi, ako ne pronađemo vrednost izvršava se grupa naredbi iza OTHERWISE

FOR STEP NEXT je petlja koju ne bi trebalo posebno komentarisati da nema jedne bitne razlike u odnosu na bejzik kada u bejziku napišete FOR I=100 TO 0, sadržaj petlje će se izvršiti jednom a onda će nailšavši na NEXT I računar izaći iz petlje primetivši da nije trebao ni da je izvršava. U komalu je stvar daleko pravilnije FOR i =100 TO 1 DO neće biti izvršeno nijednom, računar će odmah preći na izvršavanja programa iza NEXT I

REPEAT UNTIL I WHILE END WHILE petlje su nađemo se poznate svima koji su se upoznali sa osnovama paskala, a ostali mogu da nauče sve što je potrebno čitajući naš napis „Sa bejzika na Paskal" iz prošlih „Računara", ili gledajući primer sa slike 1

Zanimljivo je da će komal interpreter, pre nego što započne izvršavanja programa, prijaviti sve greške koje se tiču nepropisno konstruisanih kontrolnih struktura, pogledajte na primer dijalog sa slike 2 u kome se koristi i postojeće naredba GOTO iza koje se, umesto broja linije, navodi ime labele



Procedure i funkcija

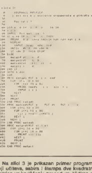
Autor ovoga teksta mora da prizna da je uvek bio zadovoljan običnim IF THEN ELSE i FOR NEXT i da je glavni praktičan napredak Komala očekivao na polju rada sa procedurama i funkcijama. Nije nam, naravno, prevelika želja da pišemo struktuirane programe, ali je mogućnost prenošenja argumenata (posebno nizova) u potprograme neophodan preduslov za komforan rad sa matricama i ostale inženjerske proračune. Tvorci komala su na polju prenošenje argumenata ispunili sva očekivanja i uradili dosta stvari koje su i najveći optimisti teško mogli da zahtevaju

slika 4

KLJUCNE REČI COMALA

ABS	ASC	ADVAL	ADN	APPEND	ASR	ATR	AUTO	CASE
CHR$	CLEAR	CLG	CLOSE	CLOSED	CLS	COLOUR	CORT	COS
COUNT	DATA	DEBUG	DEG	DEL	DELETE	DIM	DIV	DO
DRAW	EDIT	ELIF	ELSE	END	ENDCASE	ENDFUNC	END IF	END WHILE
END PROC		ENVELOPE		EOD	EOF	EOR	EXEC	EXP
EXT	FALSE	FILE	FON	FREE	FUNC	GCOL	GET	GET$
GOTO	IF	IMPORT	IN	INKEY	INKEY$	INPUT	INT	LEN
LIST	LR	LOAD	LOG	MOD	MODE	MOVE	NEW	NEXT
NOT	NULL	OF	OLD	OPEN	ON	ORD	OSCLI	OTHERWISE
PAGE	PI	PLOT	POINT	POS	PRINT	PROC	RAD	RANDOM
READ	READ ONLY		REF	RENUMBER		REPEAT	RESTORE	RETURN
RND	RUR	SAVE	SELECT OUTPUT		SGN	SIN	SIZE	SOUND
SQR	STEP	STOP	STR$	TAB	TAR	THEN	TIME	TO
TRUE	UNTIL	USING	USR	VAL	VDU	VPOS	WHEN	WHILE
WIDTH	WRITE	ZONE						

Napomena: ovo je spisak naredbi komala pisanog za BBC B i Electron koji se, u ROM-u od 16 kilobajta, može nabaviti od Acornsoft Limited, Betjeman House, 104 Hiles Road, Cambridge CB2 1LQ za 43 funte+VAT.



Na slici 3 je prikazan primer programa koji učitava, sabira i štampa dve kvadratne matrice uz korišćenje procedura. Vidimo da se procedura pozivaju prostim navođenjem njihovog imena i spiska argumenata, dok se njihove definicije upisuju na kraj programa uz korišćenje službene reči PROC (FUNC za funkcije). Argumenti se mogu prenositi po imenu i po vrednosti. Neka smo, na primer, napisali *proba (a, b, c)* a docnije PROC *proba (p, q, r)*. Pozivanjem procedure formiraju se promenljive p q i r koje dobijaju vrednosti promenljivih a, b i c. Po završetku rada procedura promenljive p, q i r će jednostavno biti zaboravljena, dok će a, b i c imati vrednosti kao i pre poziva. Na ovaj način, u proceduri nije moguće promeniti vrednost stvarnih argumenata

Da smo napisali PROC proba (REF p,REF q r) situacije bi bila drukčija a i b su promenljive pranete **po imenu** i njihova vrednost sme da se promeni u proceduri, dok će c, preneto **po vrednosti**, biti nepromenjeno delovanjem ovako poznatog potprograma.

Promenljive iz glavnog programa su na raspolaganju u procedurama koje smeju da ih menjeju kao takozvane *globalne promenljive*. Ukoliko ne želimo da dopustimo potprogramu da menja vrednosti iz glavnog programa, deklarisaćemo proceduru kao CLOSED. Ukoliko nam je potrebno da u proceduri koristimo samo neke od promenljivih iz glavnog programa „uvešćemo" ih u proceduru putem deklaracije IMPORT. Samo se po sebi razume da argumenti, lokalne i globalne promenljive procedura mogu da budu kako celobrojne, racionalne i alfanumeričke promenljive, tako i matrice, odnosno matrice stringova.

Sve u svemu, komal je jezik iz snova svakog pravog programera — jezik koma se posle vrlo kritičkog razmatranja može uputiti samo još jedna jedina zamerka — nedostatak naredbe ON ERROR GOTO koja bi, uostalom, mogla da se simulira malim mašinskim potprogramima (Comal je opremljen naredbom USR). Pogledajte, dakle, spisak naredbi Komal, nabavite interpreter i neka bejzik ustupi mesto boljem!

Dejan Ristanovic

Smešna strana računara

svaki početak je težak

Jednostavno ga uključi, ubaci disk i, ako se ništa ne dešava, pozovi svog prijatelja Jocu

Kada ste prvi put dobili računar, rečeno vam je da uz njega možete dobiti i razna čudesna priključke i pomagala. Niko vam međutim, nije pomenuo najvažnije pomagalo uopšte prijatelja koji je u stanju da vam pomogne kada stvari zapnu. Moj prijatelj je Joca, koji sedi tri kancelarije dalje. Bez njega, ja ne bih bio u stanju da ovo sada pišem na kompjuteru.

Prva neopaka stvar bila su uputstva za rad. Ova uputstva za kompjutere napisana su jezikom koji veoma liči na srpskohrvatski. To je jezik koji mogu da čitam, ali ne i da razumem. Da kompjutar već nije bio spakovan i ekspedovan, bio bih ga odbio pošto sam pročitao uputstvo.

Ja nisam mehaničar, a sasvim sigurno ni elektroničar. U stanju sam da kucam na mašini i mogu da okačim sliku na zid najviše jedan ili dva centimetra daleko od mesta na koma sam to želeo — dostignuće zbog koga se možda neopravdano ponosim. Ovaj kompjuter je, međutim daleko prevazilazio moje sposobnosti. Kad sam prvi put pokušao da ga uključim, ništa se nije desilo — osim jarke svetlosti, slične onoj koje se, pretpostavljam, javlja na kontrolnoj tabli nuklearnog reaktora neposredno pred katastrofu. Nije bilo čak ni odvratne kratke poruke koja najavljuje kvar (Failure) ili te šalje do davola (Go directly to jail). Pošto sam provario da je utikač na mestu — najviše što sam lično bio u stanju da uradim — kranuo sam do Jocine kancelarije.

Joca je našao da sam uključio disk jedinicu na tipično pogrešan način. Kada ga je postavio kako treba, blještava svetlost je išćezla, a mi smo bili nagrađeni najavom datuma.

„Sad će sve biti u redu," rekao je Joca i otišao.

Posle nekoliko minuta međutim, kad se ništa korisno nije desilo (datum se već zneo) ponovno sam pošao da potražim Jocu. „Još uvek se ništa ne dešava" rekao sam. „On jednostavno stoji uživajući u datumu."

„Da li si pritisnuo RETURN?"

„Naravno da nisam. Šta je to?"

Objasnio mi je i kad sam se vratio i pritisnuo RETURN stvar je kranula. Sklonio sam uputstva i seo da koristim svoju novu igračku.

Posle pet minuta bio sam sav u znoju, šestog minuta otišao sam da nađem Jocu, čija ljubaznost nije bila pokolebana, iako je već po drugi put objašnjavao razliku između ESCAPE, CONTROL i HELP tastera. „Nemoj koristiti HELP taster," rekao je. „On ti i tako neće pomoći. Ako ne ide, jednostavno dođi opet do mene."



U čemu sam pogrešio?

Obišao sam ga još desetak puta pre nego što sam napravio prekid zbog ručka pre koga sam uzeo dva dobra vinjaka da oteram očajanje. Ni oni međutim nisu pomogli.

Tokom sledeće dve nedelje Joca je uspeo da sačuva strpljenje vodeći me kroz lavirinte Format Line tastera kao i kroz Del, Ins, Append i Home. Na kraju došao je dan kad sam ušao u svoju kancelariju pravilno uključio disk jedinicu i sastavio dugu složenu kancelarijsku belešku pišući fraze i rečenice na ekranu skoro po želji koriguju-ći tekst. Trebalo mi je samo tri primerka beleške ali sam za svaki slučaj uradio četiri i prešao već bezbroj puta pređenih 37 koraka do Jocine kancelarije sa prvim primerkom. „Pogledaj!" rekao sam slavodobitno i pružio mu ga. Pažljivo ga je razmotrio. „Zar ti nisam rekao kako je to prosto."

Ohrabren preneo sam kompjuter kući. Prvi put međutim kad sam nešto kucao printer se u sredini treće strane odjednom zaustavio uz ljutito kreštanje. Joca je bio van grada pa sam pozvao drugog prijatelja.

„Verovatno si ostao bez trake" rekao mi je.

„Nisam znao da postoji neka traka."

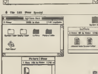
Duga pauza pre nego je odgovorio. „Kako misliš da ti bez toga štampa?" pitao je sa izvesnim čuđenjem. On nije imao Jocino iskustvo sa mnom.

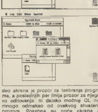
Najveća zadovoljstvo sam doživeo kad se za kompjuter zainteresovala moja žena. Trebalo je da piše mnoga službena pisma pa je pitala da li bi mogla koristiti kompjuter u tu svrhu. „Naravno" rekao sam. „To je vrlo jednostavno. Prosto ubaci disk. Pazi samo da ga postaviš kako treba."

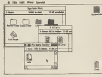
bacite prozore kroz prozor

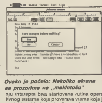
Većina ljubitelja kućnih računara je za prozore prvi put čula kada se pojavio Sinclair QL. Onda su prozori pominjani kod „mekintoša", pa kod „atarija" i, na kraju kod „amige", a čule se i tvrdnje da bi se mogli ostvariti (ili da su ostvareni?) čak i na „spektrumu". Šta bi prozori trebalo da budu, a šta oni na žalost, obično jesu?











Ovako je počelo: Nekoliko ekrana sa prozorima na „mekintošu"

Zamislite da ste srećnik koji poseduje tri računara, tri monitora i tri diska i da je čitava ta oprema povezana tako da svi kompjuteri mogu da budu uključeni istovremeno. Na prvom kompjuteru možete, dakle, da startujete jedan program, na drugom drugi, a na trećem da započnete partiju šaha. Baš zgodno!

Da li bi nešto slično moglo da se izveda na jednom jedinom kompjuteru? Moderni računari sa moćnim mikroprocesorima su obično sposobni za takozvani „multitasking" — u jednom momentu mogu da izvršavaju nekoliko programa koji su praktično nezavisni jedan od drugoga. Ako uz taj računar imamo tri monitora, na svakom od njih će biti prikazani rezultati izvršavanja jednog od programa, dok ćemo sa jedne od tri tastature moći da unosimo potrebne podatke.

Da li bi nešto slično moglo da se izvede i samo sa jednim kompjuterom, koji ima jednu tastaturu i jedan monitor? Moglo bi: dovoljno je da se ekran podeli na tri fiksna dela (prozora) i da se svakom od programa koji se izvršavaju dodeli po jedan od njih. Sve PRINT naredbe tog programa će ispisivati podatke isključivo u njegov prozor, dok će svaki INPUT očekivati da se tražena veličina otkuca dok se kurzor nalazi u istom prozoru. Korisnik, naravno, može da pomera komandni kurzor u bilo koji od prozora i tako kontroliše program koma je taj prozor dodeljen.

Bitno je da programi koji rade sa prozorima budu propisno pisani: ne sme se direktno „poukovati" u video memoriju, „pikovati" tastatura i, uopšte, koristiti bilo šta osim dokumentovanih rutina operativnog sistema. Jasno je i zašto — operativni sistem obično ne može da kontroliše POKE naredbe, pa će jedan program upisivati podatke u prostor dodeljen drugome, a ne treba mnogo mašte da se zamisli kako će se sve to završiti. Prozori će vam, dakle, pomoći da izvršavate nekoliko „ozbiljnih" programa, ali nećete smeti ni da pokušate da igrate tri brze komercijalne igre ili da se zabavljate simultankom protiv računara na tri šahovske table.

Neki kućni računari skromnijih cena imaju daleko jednostavnije rešenje prozora: ekran možete da podelite na nekoliko delova, s tim da se u svakome od njih vrši nezavisno pomeranje teksta, što znači da ćete, na primer, moći da ispravljate program u jednom od prozora i da ga testirate u drugome. Najjednostavniji vid prozora je implementiran kod „spektruma": gornji deo ekrana je prozor za testiranje programa, a poslednjih par linija prozor za njegovo editovanje, ni daleko moćniji QL nije mnogo odmakao od ovakvog shvatanja prozora. Posebna su priča ekrana za grafiku kojima ćemo se pozabaviti u nekom idućem napisu.

Ostalo je još da vidimo kako kućni računar može da izvršava nekoliko programa odjednom. Mikroprocesor, ma koliko bio moćan, može u jednom trenutku da izvršava samo jedan program. Operativni sistem, međutim, podešava hardver tako da generiše interapte u pravilnim intervalima, na primer 100 puta u sekundi. Po generisanju interapta biva startovana rutina operativnog sistema koja proverava vreme koje je proteklo od kako se izvršava neki program. Ukoliko je programu istekao dodeljeni tzv. *kvant vremena*, on biva prekinut, sadržaji svih registara upisani u memoriju, a zatim se nastavlja izvršavanje nekog drugog programa kome se dodeljuje novi kvant vremena. Ukoliko se programi razmenjuju dovoljno brzo, korisnik će imati utisak da se svi oni izvršavaju istovremeno, pri čemu ima priliku da važnom programu produži kvant vremena i tako učini da on bude brže završen.

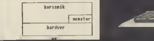
Moramo, na kraju, da kažemo da je multitasking kod personalnih računara stvar koja se upravo razvija i koja je neobično podložna bagovima: prozori se često mešaju, tekst iz jednog nepredviđeno prelazi u drugi, rad jednog programa onemogućava nastavljanje drugoga i tako dalje. Obzirom da je veliko pitanje da li je nekome uopšte potrebno da izvršava pet programa odjednom, ovim tehnikama konstruktori računara posvećuju onoliko brige koliko je potrebno da se spisak karakteristika kompjutera dopuni sa par novih redova, što znači da ćemo još pričekati da vidimo prozore realizovane na sasvim korektan način.

Dejan Ristanović

Operativni sistemi

prvi gem za tripos

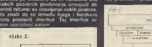
Pre nego što se pozabavimo konkretnim operativnim sistemima, neće biti zgoreg da objasnimo šta je to operativni sistem i zašto je on uopšte potreban. Prvi popularni elektronski računari kao što je SIM (simpatična stara mašina zasnovana na 6502) praktično nisu ni imali operativni sistem neki minimalni monitor je kontrolisao rad heksadekadne tastature, a na korisnika je padao teret upravljanje ulazom i izlazom — ako je, na primer, trebalo uključiti zvučnik korisnik je morao samostalno da asemblira instrukcije LDA # &FF: STA &FFFD da ih unese u memoriju i startuje tako dobijeni program. Interakcija korisnika i nekog od prvih računara je prikazana na slici 1.





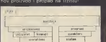
interfejs čovek-računar

Jasno je da se od korisnika koji želi samostalno da kontroliše hardver zahteva dosta znanja, iskustva i napora. Cilj razvoja računara je, međutim, da se korisniku bez ikakvih posebnih predznanja omogući da koristi računar za obavljanje nekih poslova, što znači da se između njega i hardvera mora postaviti *interfejs*. Taj interfejs se zove *operativni sistem*.

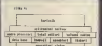


Slika 2 ilustruje konstrukciju operativnih sistema popularnih kućnih računara iz sedamdesetih i ranih osamdesetih godina. operativni sistem kombinovan sa bejzik interpretatorom se postavlja tako da hardveru tumači potrebe korisnika. Korisniku je i dalje ostavljen „prolaz" kroz koji direktno može da kontroliše inteligentne čipove svog kompjutera i to obično naredbama PEEK POKE i USR, iako se pretpostavlja da će te naredbe koristiti samo relativno mali broj ljudi. Izuzetak od ovoga pravila su napravili konstruktori računara TI 99/4A, koji su uložili veliki trud da korisniku onemoguće prilaz hardveru, zbog čega je njihov proizvod i propao na tržištu!

Nova koncepcija u komunikaciji s korisnikom: „Mekintoš"



Konstruktori novih računara kao što su šesnaestobitni IBM PC, „mekintoš", „atari 520 ST" i osmobitni BBC su pristupili stvarima kao na slici 3. Hardveru je neposredno nađeđen operativni sistem koga, za svoje potrebe aktiviraju interpretatori i kompajleri raznih jezika (bejzik, paskal, asembler ...). U čemu je prednost ovakve koncepcije? Ako operativni sistem poseduje naredbe MOVE i DRAW neće biti potrebno da se razvijaju novi algoritmi za crtanje linija koje će koristiti razni programski jezici: bejzik interpreter, paskal kompajler i korisnik koji piše asemblerska programe će, kada im zatreba crtanje linije, jednostavno pozvati potprogram operativnog sistema! Posebno je zanimljivo da će bejzik interpreter, paskal kompajler i bilo koji propisno napisani mašinski program moći da se izvrše na bilo kojoj drugoj mašini koja poseduje isti operativni sistem! Korisnik je pozvao program koji crta liniju i ne razmišljajući o grafičkoj rezoluciji, koordinatama ekrana ili njegovoj poziciji u memorijskoj mapi. Mogućnost prenošenja programa sa jedne na drugu mašinu i iz jedne u drugu generaciju mašine je, u doba rastuće cene softvera, karakteristika koja može da odluči sudbinu svakog računara.



Na slici 4 je prikazana hardversko-softverska organizacija velikih kompjuterskih sistema kod kojih korisnik nema nikakvog pristupa hardveru!

Operativni sistemi na 68000

Mikroprocesori iz Motoroline familije 68000 su pokupili mnogo komplimenata u raznim prikazima, ugrađeni su u mnoge moderne personalne računare ali, začudo, još nisu dobili standardni operativni sistem! dok se od standardnog operativnog sistema očekuje da bude „javno vlasništvo" i da se primenjuje na mašinama raznih proizvođača operativni sistemi QL-a „mekintoša" i „atarija" su zaštićeni posed ovih firmi! Pre

Čitaoci prikaza personalnih računara su već dobro naučili vladajući redosled stvari: najpre se govori o hardveru, a zatim o bejzik interpreteru. Došlo je, čini se, vreme da se kriterijumi za vrednovanje kućnih kompjutera iz temelja promene i približe kriterijumima za vrednovanje velikih sistema: umesto o bejzik interpreteru nekog kompjutera, sve češće govorimo o njegovom operativnom sistemu! GEM i tripos, operativni sistemi „atarija 520 ST" i „amige", nalaze se poslednjih meseci u centru pažnje ljubitelja računara. Iako se radi o novim i nedovoljno proverenim stvarima, već je sada potpuno jasno da je tripos u utakmici sa Tramijalovim Gem-om dobio, u najmanju ruku, svoj prvi gem.

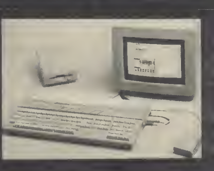
nego što kažemo nekoliko reči o posledicama ovakve situacije, pozabavićemo se operativnim sistemima koje koriste računari u koje je ugrađen MC 68000

Apple-ov Liza/makintoš OS se hronološki prvi pojavio Zamišljen je izuzetno ambiciozno skup programa koji će moći da primeni svako ko ume da pritisne običan taster! Nedovoljno je, međutim, prilagođen potrebama iskusnijih korisnika ne omogućava simultano izvršavanje više programa i, kao vlasništvo firme Apple, nema šanse da stekne popularnost na modelima drugih proizvođača

QDOS, operativni sistem koji koristi Sinclairov QL je najšire rasprostranjen na tržištu, svakako zbog veoma niske cene ovoga računara (svega 200 funti) QDOS je, međutim, orijentisan prema implementaciji bejzika u tolikoj meri da ga je teško uvrstiti u prave operativne sisteme, koji, kao što smo rekli, moraju da budu potpuno nezavisni od programskih jezika i drugih aplikacija Obzirom na činjenicu da QDOS operiše sa mikrodrajvima i da se nalazi pod strogim kopirajtom firme Sinclair Research, vrlo je teško očekivati da će se on ikada pojaviti na bilo kojoj drugoj mašini i postići širu popularnost.

UCSD-ov p sistem kod nas relativno nepoznat, nije vlasništvo proizvođača računara, ali nije ni pravi operativni sistem — radi se pre o grupi kompajlera za prevođenja programa sa fortrana, paskala i C-a na takozvani *p kod* i run time interpretera koji taj *p kod* izvršavaju Potpuno izolovan od grafike, UCSD ima budućnost kao dopuna nekom operativnom sistemu, ali je bespredmetno razmatrati ga kao samostalnu tvorevinu.

CP/M 68 K, vlasništvo čuvene firme Digital Research je uz minimalne izmene, usvojen kao operativni sistem „atarija 520 ST" Za takvu odluku konstruktori „atarija 520 ST" imaju dobro pokriće — CP/M, operativni sistem razvijen za računare sa Z80, uživa ogroman ugled i popularnost, a može da se podiči i ogromnom softverskom podrškom Podstaknuta njegovim uspehom, firma Digital ga uporno prilagođava svim mikroprocesorima koji se pojavljuju na tržištu, što se sve više pokazuje kao dvosekli mač: između osmobitnog i šesnaestobitnog postoje ogromna razlika — ono što je za računare sa Z80 bilo dobro i moćno izgleda prilično otužno kada se uporedi sa onima što može da uradi MC 68000! Pravilno procenjujući ovakvu situaciju IBM je odlučio da svoj PC ne opremi CP/M-om (premda se ta mogućnost dugo razmatrala), već da razvija sopstveni operativni sistem MS DOS — danas vladajući svetski standard na mašinama koje koriste Intelove mikroprocesore. Digital Research je, sa svoje strane, uvideo da CP/M ili bilo



„Mekintoš" za siromašne: „Atari 520 ST"

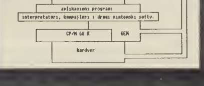
koji operativni sistem koji ne podržava grafiku ne može da se nada komercijalnom uspehu, što je dovelo do proizvodnje GEM-a — operativnog sistema za kompjutersku grafiku i komunikaciju sa korisnikom

Programska baterija GEM

Na slici 5 je prikazana interakcija GEM-a sa CP/M-om i hardverom GEM, kao što vidimo, nije operativni sistem, jer se dobrim delom oslanja na CP/M koji mora da bude prisutan u mašini Sa druge strane, GEM jeste operativni sistem zato što direktno kontroliše deo hardvera računara i zato što se drugi aplikacioni programi mogu pozivati na njega. Obzirom da je u ovom trenutku implementiran na „atariju 520 ST" IBM-u PC/AT i mnogim kompatibilnim mašinama GEM možemo smatrati grafičkim operativnim sistemom koji ima velike šanse za tržišni uspeh

GEM je koncipiran tako da bude nezavisan od uređaja na koji će upućivati rezultate svoga rada U „normalnoj" situaciji se GEM, jasno, koristi za crtanje po ekranu, ali se jednom jedinom naredbom može definisati printer, ploter ili neki specijalni grafički terminal kao odredište slike GEM je dakle, prvi operativni sistem od koga možete da tražite da nacrta krug na matričnom štampaču, a da pritom ne promeni sadržaj ekrana!

Osim naredbi za crtanje pravih linija, mnogouglaonika i kosih elipsi (specijalan slučaj kose elipse je, jasno, i krug), GEM je opremljen naredbama za bojenje zatvorenih površina koje koriste izuzetno efikasan i brz upravo patentirani algoritam, kao i



opcijama za rotiranje i zumiranje čitave slike i njenih delova. Ugrađivanje grafičkih naredbi koje pozivaju GEM u bejzik ili korišćenje BCPL-a, specijalnog jezika za kompjutersku grafiku, može dovesti do spektakularnih rezultata.

Običnog korisnika će, međutim, daleko više interesovati baterija programa GEM Write, GEM Paint i GEM Draw koje će dobiti uz GEM. *GEM Write* je, naravno, tekst procesor solidnih karakteristika rađen po ugledu na MacWrite, tj. tako da ga mogu koristiti i ljudi nevični radu sa računarom. Posebno je značajno da se u tekstove koje pišete na GEM Write-u mogu lako uklapati crteži dobijeni primenom GEM Paint-a i GEM Draw-a.

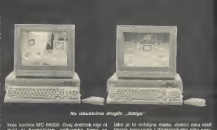
GEM Paint je program koji omogućava crtanje po ekranu računara i to u 16 boja uz obaveznu primenu naredbi ZOOM (uvećanje nekog dela slike), CUT i PASTE („sečenje" delova slike i njihovo „lepljenje" na neko drugo mesto). Iako se u „Računarima" do sada nismo bavili programima za crtanje ili CAD (Computer Aided Design), reći ćemo da GEM Paint spada u takozvane *freehand* programe, tj. programe koji bi trebali da omoguće i crtanje neke vrste umetničkih slika. Za razliku od njega, većina programa za crtanje se zasniva isključivo na pravim linijama i polukrugovima, što je sasvim dovoljno za većinu inženjerskih primena. Zato je, verovali ili ne, jedna od ekskluzivnih primena GEM Paint-a crtanje stripova, karikatura i stilizovanih ilustracija!

GEM Draw je program za crtanje grafika, blok dijagrama, tabela, histograma i „pie chart" dijagrama i kao takav predstavlja izuzetno korisnu alatku za svakoga kome je potrebna bilo kakva tehnička dokumentacija.

Sve u svemu, GEM je izvanredan grafički operativni sistem koji dopunjava jedan od najvećih nedostataka CP/M-a 68 K. Na žalost, CP/M, posmatran sa aspekta šesnaestobitnih mašina, ima i mnoge druge nedostatke koje bi trebalo popraviti dodavanjem novih operativnih sistema koji, naravno, povećavaju cenu računara i u najvećem broju slučajeva otežavaju njegovu primenu. Zbog toga je, možda, bolje sastaviti opšti operativni sistem velikih mogućnosti, baš kao što je Metacomco uradio za „amigu".

Avangarda zvana tripos

Metacomco je u stručnim krugovima prilično poznata softverska firma sa sedištem u Bristolu (Engleska). Zadovoljan radom ove firme na realizaciji grupe uslužnih programa za QL-a Commodore je od Metacomca zahtevao da radi na „rezervnom" operativnom sistemu za „amigu", ovaj bi se operativni sistem primenio samo ukoliko Commodorovi inženjeri ne uspeju da razviju sopstveni sistemski softver. Prva demonstracija tada još nedovršenog triposa je učinila da Commodore u trenutku odustane od razvoja koji je samostalno započeo i ugradi tripos u „amigu" — računar od koga svakako mnogo očekuje. Metacomco je, sa svoje strane, odbio da proda tripos Commodoru i, u relativno kratkim pregovorima, uspeo da izbori pravo da svoje remek-delo implementira i na drugim mašinama



Na iskustvima drugih: „Amiga"

koje koriste MC 68000. Ovaj dobitak nije ni mali ni beznačajan, softverska firma se uvek trudi da implementira svoj operativni sistem na što većem broju računara, jer tako dobija više para. Proizvođači računara su, nasuprot tome, često skloni da dobre operativne sisteme zadrže za sebe i tako smanje konkurenciju.

Tripos je, da počnemo od samog početka, dobio ime po tronošcu na kome su studenti Kembridža sedeli dok su, pre par stotina godina, polagali ispite. Docnije je reč tripos korišćena kao šaljivi naziv za trostepene studije na istom univerzitetu, da bi, uvažavajući činjenici da se završava sa OS, postala ime modernog operativnog sistema.

Tripos je operativni sistem koji, bez ikakvih trikova, omogućava istovremeno izvršavanje velikog broja programa. Svakom programu (bolje bi bilo reći procesu) korisnik dodeljuje prioritet, a računar se brine o tome da procesi nižeg prioriteta budu startovani tek kada prioritetniji čekaju na ulaz ili izlaz podataka. Osim ovo, tripos ima još tri suštinske karakteristike: flopi diskovi ne koriste sektore već čitave trake, ne postoji traka u kojoj se nalazi direktorijum diskete i ne postoje ograničenja praktično ni za šta!

Za svaku otvorenu datoteku na disku operativni sistem rezerviše po jedan bafer, tako da WRITE naredbe nemaju potrebe da startuju disk — podaci se upućuju u bafer što je daleko brže. Bafer se, jasno, brzo puni pa ga treba prazniti upućivanjem podataka na disk. To radi jedan od stalno prisutnih procesa niskog prioriteta — kad god nema šta da radi, „amiga" prepisuje podatke iz bafera na disk. Ukoliko se dogodi da se neki bafer prepuni, proces koji ga je punio će biti prekinut i ustupiće procesor nekom drugom programu da bi bio ponovo aktiviran kada u baferu bude bilo mesta. Ovakva koncepcija, pozajmljena od velikih kompjuterskih sistema, ima jednu veliku manu — u slučaju nestanka struje datoteke će biti nepropisno zatvorene, pa je moguć gubitak dela podataka. Dobrih strana je daleko više — zamislite da ste izdali komandu SAVE i da je računar ispisao prompt koji vam omogućava dalji rad a da se disk nije praktično ni zavrteo!

Odluka da se eliminiše traka na koju bi bio upisan direktorijum diskete je, na prvi pogled, prilično neobična — navikli smo da komanda DIR na ekranu u trenutku ispiše imena svih snimljenih programa. Amiga će, kada otkucamo DIR, postepeno pretraživati disketu prateći stablo poddirektorijuma što će eventualno potrajati nekoliko sekundi. Iako je to ozbiljna mana, dobici nisu mali: imena programa i direktorijuma nisu ograničena na sedam ili deset slova što znači da ćemo moći da napišemo SAVE „NULE FUNKCIJA" a ne da nazovemo program NUFN i da se docnije pitamo šta on radi. Datoteke, osim toga, mogu da budu dugačka onoliko koliko je potrebno i da se prostiru preko nekoliko disketa što je vrlo teško ostvarljivo na većini konkurentnih operativnih sistema.

Operisanje se trakama a ne sa sektorima omogućava ne samo povećanje brzine rada nego i smeštanje 880 kilobajta informacija na disketu od 3.5 inča. Programeri koji žele da štite svoje programe su verovatno jedini koji se neće obradovati ovakvoj organizaciji: u međusektorske razmake se obično upisuju razni šifrovani podaci koji otežavaju neovlašćeno kopiranje programa, dok će za „amigu" ti podaci biti standardne informacije koje se bez problema kopiraju!

U okviru prikaza „amige" koji objavljujemo u ovom broju „Računara" se daleko kompletnije bavimo grafičkim karakteristikama triposa i njegovom interakcijom sa korisnikom, ovde nam je dovoljno da kažemo da je tripos operativni sistem iz snova za svakog Pravog Programera i da ima sve šanse da postane važeći standard na računarima koji koriste Motoroline mikroprocesora.

Na putu triposovog uspeha može da se isprači jedino UNIX, moćni operativni sistem pisan za velike kompjuterske sisteme koji je već implementiran na Motoroli 68000. Iako ćemo se karakteristikama Unix-a baviti nekom drugom prilikom, vredi pomenuti da se radi o operativnom sistemu koji su programeri pisali za programere, izuzetno kompleksnom, moćnom i fleksibilnom. Unix podržava ne samo istovremeno izvršavanje više programa nego i istovremeni rad velikog broja korisnika i može da se pohvali moćnom softverskom podrškom — verovatno većom nego CP/M i svakako neuporedivo većom nego tripos. Mana Unix-a je što može da se instalira jedino na skupom hardveru: hard disk od 5 Mb je praktično neophodan čak i za najskromniju verziju Unix-a, dok se komforan rad postiže tek sa diskovima od 20 megabajta. Moguće je da će pad cena masivnih diskova učiniti Unix vladajućim programerskim standardom, ali će u međuvremenu tripos biti na samom vrhu piramide; moguće je da Metacomco za devedesete godine bude ono što je Digital bio za sedamdesete, a Microsoft za osamdesete.

Dejan Ristanović



Računari iz mog ugla



samo preko nas mrtvih

Ko kaže da se kod nas ništa nije uradilo po pitanju uvođenja računara u škole? Baš nam je bilo lepo dok smo držali savetovanja simpozijume sednice sastanke okrugle stolove, tribine To je urodilo plodom pa smo dobili rezultate — planove, programe, analize koncepte manifeste reforme, zahvete

To su sve same krasne stvari, do suza vas može dirnuti briga nadležnih za budućnost mladih pokolenja Citiram neznanog junaka „U našim krajevima ima malih takoreći patuljastih škola, koje nemaju ni tablu, ni kredu pa im onda ne treba ni računar" Tečno patuljci treba da rade u rudniku, a ne sa računarom Ima i predloga da ne uvodimo računare ni u druge škole, da se među učenicima ne bi pravile socijalne razlike

Samo nemojmo žuriti Oprezno! Korak po korak, ali stabilno! Polako ali sigurno Neobično nas je obradovala činjenica da ni Nemci, navodno, nisu prenaglili i uveli kompjutere u škole

Eto prilike da se primene dva univerzalna filozofska grifa, koji su kao stvoreni za prosvetne radnike umerenost i relativizam Te divne osobine prenose se i na mlada naraštaje koji su na dobrom putu da postanu umereno-relativni ili relativno-umereni

Umerenost — to je malopre pomenuta opreznost ne zaletati se, čekati To se potkrepljuje divnim narodnim poslovicama „Sve što je brzo je kuso" ili „Triput meri — jednom seci" Nema sumnje da je narodni umotvorac, smišljajući ove bisere narodne mudrosti, imao na umu baš uvođenje računara u škole Ali ako to premeravanje potraje može da nam pobegne trajvan za 21 vek

Drugi zahvat relativizam, sadržan je u rečenicama sa „ali" Na primer „Računari su bez sumnje vrlo potrebni u školi, ALI " Sami možete završiti rečenicu Tako ćete u svakoj prilici izgledati pametno

Samo zlobnici mogu da tvrde da se ne radi o opasnosti po decu nego po profesore i da se ovi plaše da će izgledati smešno u ambijentu novog vremena

Do sada je deci utenvana trauma po pitanju matematike a od sada postoji opasnost da deca uteruju traumu profesorima uz pomoć kompjutera

Sirota nastavnica matematike smrzava se od pomisli na svu tu elektroniku u računaru kad još nije došla do stadijuma da ume da zameni osigurač u svom stanu To još i nije najstrašnije nego kad vidi sa koliko se malo respekta i kompleksa klinci upuštaju u odnose sa računarom Oni možda ne znaju mnogo ali nije ih sramote da probaju i ne uspeju Ona sebi ne može dozvoliti takav rizik da se pred svima saplete Mogla bi to da proba kod kuće ako ima računar ali u tom slučaju neće stići da ode na pijacu i skuva ručak Klinci mogu mnogo toga što ona, štiteći svoj autoritet sebi ne može da dozvoli i da se igraju i da pitaju gluposti i uopšte ti novi klinci nekako bezobrazno razmišljaju i postavljaju drska pitanja Oni se osećaju komotno uz računar To je sredstvo koje ubrzava vreme na neki čudan način ne samo da se nema vremena za odlazak na pijacu i kuvanje ručka, nego se i efekat dilatacije vremena ogleda i u tome što profesori zaostaju čak brže od računara

Do sada nam je uspevalo da držimo računar na distanci ali to više nije moguće Neprijatelj se uvukao u naše redove i predstoji nam žestok obračun borićemo se do poslednjeg učenika Među prosvetnim radnicima još nema panike, ali se oseća izvesna nervoza Nema razloga za preteranu uznemirenost Postoji čitav niz mogućnosti za ostvarenje vrhunske ideje dobre kako deci ogaditi računare

Za postizanje tog plemenitog cilja profesorima stoji na raspolaganju više proverenih, suptilno-pedagoških metoda i sredstava

Didaktički je neopravdano predavati deci ono što bi volela da uče Profesor treba da predaje ono o čemu najbolje zna da priča To je jedan od najboljih načina zaštite od neprijatnih pitanja na koja ne ume da odgovori

Sledeća dobra metoda je uopštavanje Deci treba servirati masu uopštenih informacija o računaru koje se ne mogu neposredno iskoristiti To se zove opšte obrazovanje Da bi se otkrilo gde bi se ta saznanja mogla iskoristiti potrebno je okrenuti još jedan krug Većina nema vremena za još jedan krug, oni ulete u ozbiljne vode i tako tragično okončaju svoj razvoj Ovu metodu treba svojski primenjivati kod vaspitavanja naučnog podmlatka Jako je važno da im sopstvenim primerom pokažete kako treba da ih kao gospod Bog oblikujete prema svom liku, i to tako što ćete im predavati isključivo ono što im nikad u životu neće biti potrebno Budući naučnici treba da vežbaju da se bave stvarima koje veze nemaju sa životom Jedino tako će biti sigurni da ih praksa nikada neće demantovati

Nema potrebe prilagođavati školu vremenu u kojem se živi jer škola ne postoji zbog učenika nego zbog profesora Dok se nisu istrošili odgovarajući profesori dotle je bilo i latinskog u školama Slično bi trebalo postupati i kod izbora programskog jezika Idealan bi bio neki klasičan jezik kao ALGOL, za koga su svi čuli ali ga niko ne koristi Trebalo bi se intenzivnije služiti i nekim predklasičnim ostvarenjima neprolazne vrednosti kao što su aritmometri, tabulatori i logaritmske tablice

Kod izbora školskog računara takođe treba biti vrlo obazriv i pokazati razumevanja za više interese

Da bi se sve opštine ravnomerno i kompletno razvijale potrebno je da svaka proizvodi svoj računar koji će biti apsolutno nekompatibilan sa računarima drugih opština Shodno tome treba prilagoditi i nastavne planove Tako će se ostvariti svi preduslovi za svojevrsni prosvetno-privredni perpetuum mobile Industrija će da daje pare za školstvo a školstvo će da finansira domaću industriju računara, kupujući od njih kompjutere koje niko neće

Kada dođe vreme, možda već 2000-te godine svi ćemo se opredeliti za isti kompjuter Tada će se ostvariti optimalni uslovi u školama — za svakog učenika po jedan računar Neizvesno je samo da li će to biti galaksija spektrum ili nešto još savršenije

Jelena Rupnik

mali oglasi

SPEKTRUM

Mc SOFTWAREI SPEKTRUMOVCI!
Najbolji i najnoviji programi u kompletima od 14 do 18 programa za samo 700 din +kaseta. Rok isporuke 1 dan Besplatan katalog Komplet 20 Frenkis One on one C5 Olive, Night shade (Ultimate), Death star interceptor, Herbert s dammy run Rocky — boxing, Celis Teles ot srebian Roland's rai race Hyper sports Jewels of Babylon Buck Rogers Komplet 22 Buyle bitten, Video pool On the run Out of shadows Konoc-koul 1 2 Find gold Komplex Stenaly Royal birkdale mighty magus Dam busters (US Gold) Battle for midway Highway encounter Komplet 23 Frank Bruno's boxing 1—4 (4 borbe), Cold greny Formule 1 (CRL) Helichopper Squash Nicotine nightmare Nautilus Boller house Cyboiron Destroyer Avizyn Dodo (blaby) Twilight zone Beraldrop Komplet 18 SP4 Vs SP3 Chuckie Egg 2 Spy hunter
Milošević Zoran Pere Todorovića 10/38 11030 Beograd tel 011/552-895

Spektrum rainbow softvare vam nudi izbor od preko 15000 najjeftinijih pakal programa Kompleti od 25 programa 800 dinara Katalog besplatan Mihajlovski Kiročo Moša Pijade 128 91300 Kumanovo tel 091/23-800

SPEKTRUM UBEDLJIVO NAJJEFTINIJI PAKETI NA YU TRŽIŠTU ZAHTEVAJTE NOV BESPLATAN KATALOG SA PREKO 1000 PROGRAMA
Radulović Rade Vožatski put 10 61000 Ljubljana tel 061/225-5888

PRILIKA KOJA SE NE PROPUŠTA
BOGAT IZBOR
početnica, popularnih i stručnih knjiga na engleskom jeziku
PO NABAVNIM CENAMA
mikroprocesori
programski jezici
programiranje
operativni sistemi
spektrum
komodor 64
BBC
QL
amstrad
apl II
KNJIGE SU POTPUNO NOVE RASPOLAŽEMO SAMO PO JEDNIM PRIMERKOM
Milovanović Ljubiša tel 011/558-007

• Sve za ZX spectrum
— video i logičke igre
— namenski programi
— uputstva za programe
— knjige i prevodi
— veliki izbor za Commodor BBC i QL
Garancija za sve vrste usluga Beta Basic 3 0 direktno iz Londona sa kompletnim original uputstvima možete dobiti samo kod COMET SOFT-a!
Milovanović Ljubiša Petra Lekovića 57 11030 Beograd tel 011/558-007

Komodor

COMMODORE 64 — NAJKVALITETNIJI PROGRAMI I LITERATURA U JUGOSLAVIJI NAJNOVIJE IGRE GARANCIJA KVALITETA I PRAVO NA REKLAMACIJU BESPLATAN KATALOG SA OPISIMA PROGRAMA
Nenad Radosavljević Cvijete Zuzorić 39 41000 Zagreb

Commodore 64 — najnoviji (hit) kasetni i disk programi! (Winter games elite, Beach-head II itd)
Čurčić Dušan Osmana Đikića 16/A 11000 Beograd tel 011/762-022

Commodore 64 — specijalni programi za ubrzano presnimavanje programa sa diska na traku! Svaki od njih vam donosi značajnu uštedu (50%) u vremenu! Max dužina programa koji se kopiraju je 193-199b Pizza turbo fast 3000 din pizza turbo II 4000 d pizza turbo III 5000 d Fast modul turbo 5000 d GTX protect copy 4000 d GTX normal copy 5000 d Štedite svoje vreme! Hit program – The music studio sa kompletnim uputstvom i na disketi 4500 d
Sve informacije na tel OLL/Mirko Žagar Vukasovićeva 82 11090 Beograd

Hit igre za C-64! Najbolji tabrički programi za malo novca Uverite se u brzinu i kvalitet isporuke Tražite besplatan katalog M&S Software III Bulevar 130/193 11070 N Beograd Tel 011/146-744

Prodajem kompletan sistem mini računara komodor 64 (računar disk drajv štampač i dve palice za igre) takođe prodajem gramofon Rolet stereo kolutni magnetofon Aris tv-igre Bush Sve ocarinjeno
Kupujem stereo kasetofon (deck)
Etramov Momčilo Volgogradska 4 III/10 Skopje tel 255 435 posle podne

GALAKSIJA

Prodajem povoljno računar galaksije (4+8 KB) posebno komplet tastaturu sa konektorom Kupujem širokokutni objektiv i teleobjektiv sa „Zenit" Ostojić Nedeljko 54000 Osijek B Kidriča 27

MEČ ZA TITULU SVJETSKOG PRVAKA U ŠAHU NA KASETI ZA KOMODOR 64 ODMAH PO OKONČANJU MEČA CIJENA 2000 DINARA+KASETA POUZEĆEM
D Bečić Obala v Stepe 24 71000 Sarajevo

QL

QL SOFTWARE — VELIKI IZBOR LITERATURE (BCPL FORTH TOOLKIT ASSEMBLERI MONITORI IGRE KNJIGE) TRAŽITE BESPLATAN KATALOG SA NAZNAKOM ZA „QL"
Nenad Radosavljević Cvijete Zuzorić 39 41000 Zagreb

RAZNO

Prodajem štampač Seikosha GP100A direktno priključiv na računare sa Centronics interfejsom (amstrad BBC spl I) uz interfejs na „spektrum" i komodore cena 130 000 din Telefon 422-837 16—18 časove

P N P electronic
HARDVER ZA SPECTRUM POVOLJNO PRODAJEMO KEMPSTON JEDNOSTRUKE I DVOSTRUKE INTERFACE ZA PALICU PALICE I/O INTERFACE, A/D-D/A KONVERTORE SVJETLOSNU OLOVKU EPROM PROGRAMATOR, MEGAROM AUDIO POJAČALO PREPRAVLJENI ROM ZA SPEKTRUM PROGRAMIRAMO EPROME POPRAVLJAMO „SPEKTRUM" I KOMODOR' I JOŠ MNOGO TOGA BESPLATAN KATALOG
P N P electronic Jaretova 12 Split

„Galaksija" „spektrum" — Prodajem pločice za sve šeme do sad objavljene u časopisima izrađa na vitroplastu i pertinaksu Cena povoljna Slobodan Đorđević 18410 Doljevac

Prodajem programe za BBC („elektron", „amstrad", „komodor" i „spektrum" Besplatan katalog Niska cena isporuka u roku od 48 sati Dragan Jovanović Kruševac Dragoslava Jovanovića 12 tel 037/30-568

Prodajem džojstik — palicu za „komodor" i „spektrum" Četiri pravca četiri pravca ukoso i okidanje sa vrha palice Izvanredno oblikovana palica sa džekom za priključivanje i katalogom — 3200 din pouzećem Pozivati u večernjim časovima na tel 037/29-550
Stojković Goran Trogirski trg 2 37000 Kruševac

Prodajem štampač Stari „Džemini — 15X tel 011/781-687

Prodajem kompletan APPLE IIc računarski sistem (128 K dva disk drajva monitor printer „epson RX 80 F/T+ Softver i upoznavanje tel 011/331 753 Pantelić Dušan Kneza Miloša 17 Beograd

Prodajem računar Acorn — elektron sa interfejsima za palice i štampač tel 011/771-509

NAJBOLJA — ODABRANA LITERATURA KOMPLETNO PROFESIONALNO PREVEDENA I KVALITETNO ŠTAMPANA!!! PROVJERENI I NEOPHODNI PRIRUČNICI **COMMODORE 64** PROGRAMMER S REFERENCE GUIDE — 1300 d MAŠINSKI JEZIK ZA POČETNIKE — 1450 d ZVUK I GRAFIKA NA C-64 — 780 d, UMJETNOST GRAFIKE NA C-64 — 900 d BASIC PRIRUČNIK — 660 d SIMON's BASIC — 660 d I PASCAL — 450 d
AMSTRAD CPC 464 (SCHNEIDER). kompletno UPUTSTVO ZA RAD NA AMSTRADU — 1100 d LOCOMOTIVE BASIC — 12000 d ARHITEKTURA I OPERATIVNI SISTEM CPC 464 — 1600 d MAŠINSKO PROGRAMIRANJE ZA POČETNIKE — 1300 d I GRAFIKA I ZVUK ZA CPC 464 — 850 d
SPEKTRUM BASIC PROGRAMIRANJE I BROŠURA „UVOD" — 700 d ISPORUKA ODMAH POUZEĆEM
DUŠKO BJELOTOMIĆ, CENTAR 1 54550 VALPOVO tel 054/82-665 ili 041/683-141

Kako da vaš mali oglas izađe u „Računarima"?

Računari objavljuju sve više male oglase Sve što je potrebno da uradite je da napišete vaš mali oglas odlučite se da li želite običan mali oglas ili mali oglas u okviru uplatite novac u najbližoj pošti i sve to zajedno pošaljete na adresu redakcije („Galaksija", Bulevar vojvode Mišića 17 11000 Beograd za male oglase) Ako ste baš lenji previše zauzeti možete da pozovete redakciju na telefon 650-181 svaki dan od 10—14 časova i izdiktirate nam vaš mali oglas Mi ćemo vam onda naknadno poslati ispunjenu uplatnicu

Koliko košta mali oglas

Obični mali oglasi do dvadeset reči koštaju 400 dinara Svaka reč preko dvadesete se naplaćuje još 20 dinara Oglas ne sme da ima više od 50 reči Adresa oglašivača se ne računa u cenu

Cena malog oglasa sa okvirom se utvrđuje malo drugačije Jedan visinski centimetar uokvirenog oglasa u standardnom „Galaksijinom" stupcu košta 300 dinara s tim što se mogu zakupiti najmanje po 32 slovna znaka Ako i ne iskoristite čitav prostor u jednom redu računa se broj redova a ne broj znakova

Molba

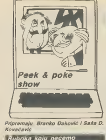
Da bi čitaoci brže shvatili suštinu vašeg malog oglasa bilo bi dobro da oglas počinje sa Prodajem Kupujem Držim časove Menjam ili nečim sličnim što ukratko ukazuje na sadržaj oglasa.

Rokovi?

Svi mali oglasi koji stignu do 1 u mesecu izaći će u broju „Računara" koji izlazi petnaest dana kasnije

Popusti

Oglasi za prodaju originalnih programa za računar „galaksija" objavljuju se besplatno pod uslovom da nemaju više od dvadeset reči Za duže oglase treba doplatiti samo razliku u ceni Na oglase za prodaju originalnih programa za sve ostale kućne računare odobravamo popust od 25% pod uslovom da oglašivač uveri redakciju u autorstvo programa



Pripremaju Branko Đaković i Saša D. Kovačević

Rubrika koju nećemo zaboraviti

Uvod

Evo baš ovih dana na godišnjicu svog postojanja i veoma uspešnog ako ne i više od toga delovanja ugasila se prva stalna rubrika o kućnim računarima alias kompjuktorima na programima talasa naše radio difuzije Naravno reč je o rubrici odnosno bloku u emisiji Ventilator 202 autora tonskog snimatelja i realizatora muzičkog saradnika i voditelja Zorana Modlija

Razrada

Dakle jednog sunčanog ali prohladnog dana ta kako svi netalentovani i nemaštoviti novinari vole da istaknu Orvelovske 1984 ? emisija Ventilator dobila je i novu rubriku o novoj modi koja je zahvatila eto i ove naše prostore Ispočetka stidljiva kratkotrajna i skromna a potom iz emisije u emisiju sve drskija bezobraznija i halapljivija (što se vreme trajanja tiče) već pomenuta i nikad dovoljno oželjena rubrika pomogla je da se mnoge stvari vezana za računara razjasne Njenom pomoći i njenim zalaganjem mnogi klinci i njihovi preci su naučili i shvatili da komodor 64 nije ime nove australijske hevi-metal grupe da „Sinkler Spektrum nije novi centrafor fudbalske reprezentacije Velsa koji je zamenio oodarenog Jana Reša i da haker nije reč u jeziku gornjomolučkih urođenika koja označava ljudavnika sestra od ujakovog strica

Znajući da na zna sve a želeći da ne samo on nego i njegovi slušaoci sve saznaju Zoran Modli uz pomoć svog nezamenljivog saradnika Duleta Pančice u svoju emisiju dovodi sve ljude koji su relevantni za kompjuktorski pokret ili bar tako izgledaju Kroz Ventilator defiluju bradati diplomirani inženjeri organizacije precizni matematičari ali i dvanaestogodišnji ili tek nešto malo stariji osnovci ponosni na svoj prvi program pod naslovom Kako pobeći sa časa istorije Svi su oni govoreći o računarima i deleći svoje i tuđe programe na presnimavanje pomogli da treća tehnološka revolucija zaluta i u naše krajeve

Zaključak

Rok je večit, a Informatika samo neprijaten ispit na fakultetu

Knjige iz Londona

Paralelno sa računarskim osvajanjem naših krajeva došlo je i do pojačanog interesovanje izdavača za oblast računara i računarske tehnike Na žalost to interesovanje je do sad uglavnom bilo koncentrisano na dve sasvim različite oblasti na uskostručne računarske knjige koje se tiču programiranja na mašinskom jeziku čipova i raznih tehničkih aspekata računara i ne računarske početnice koje pokušavaju da šarenim slikama i neodređenim pričama otprilike pokažu šta su to računari i za šta bi trebalo da služe Negde između te dve kategorije nalazi se izvestan broj knjiga posvećenih bejziku pojedinim popularnijim modelima računara i to je sve

Pojedini izdavači su pokušali da problem izbora knjiga reše tako što su uvezli određene količine knjiga engleskih izdavača (sa po kojom knjigom iz SAD i Nemačke) sa jasnom namerom da pokriju upravo ona područja na kojima su naši izdavači najtanji Na žalost takvo rešenje može biti samo privremeno i namenjeno veoma uskom krugu ljudi — ne toliko zbog engleskog jezika koliko zbog cena uvoznih knjiga Takav predah trebao bi da omogući našim izdavačima da bar malo pristignu potrebe (trenutno vlada veoma veliko interesovanje za knjige o amstradu i atariju o kojima ne postoji kod nas ni jedna knjiga) domaćih računardžija

Društvene analize računara predstavlja oblast u kojoj je najuočljiviji nedostatak publicističke podrške Na žalost umesto takvih knjiga veoma je verovatno da ćemo i dalje dobijati futurističke knjige o tome šta će sve moći računari kroz 50 godina (mi ni sad ne znamo šta sve oni mogu bar najveći deo nas) uz gotovo infantilna insistiranja da samo treba napuniti sve škole i fabrike računarima i svetla budućnost je već tu

Računari su svoje dečje bolesti izgleda preboleli ali ih zato sada boluje računarska publicistika Što pre ozdravi to bolje!

Računari na TV

Televizija zaista brzo reaguje! Izgleda da je već otkrila da postoji nešto nazvano računarski bum Shvatila je da je jedino i ispravno reagovanje uvoz Uvezena je britanska obrazovna serija The age of computers) kod nas prevedeno kao Računari') Emituje se svakog utorka u 19 časova i traje 30 minuta

Serija je pravljena 81 godine i sasvim je moguće da je TV čekala da dovoljno pojeftini pa da je dobijamo u nekom od paketa namenjenih rasparčavanju u školskom programu Na svu sreću nije mnogo propušteno Serija jeste pravljena pre četiri godine ali je u prve dve epizode uspela da nam prenese isto toliko saznanja iz sveta računara koliko i sve domaće emisije i serije krpljene i improvizovane na tu temu Ostaje da se vidi kakvi će biti sledeći nastavci ali prve dve su imale onaj sastojak koji je neophodan za jednu ovakvu seriju gotovo elementarnu jednostavnost i razložnost koja je zaista neophodna kada se o računarima govori ljudima koji su mlađi od 7 a stariji od 27 godina — znači najvećem delu TV gledališta

Pored svojih informativnih kvaliteta emisija ima i određenih mana Mane vezane za zastarelost materijala slobodno se mogu oddaciti kada se ima u vidu da je emisija pravljena 81 godine, dakle negde u srednjem računarskom veku Ono što stoji je zamerka da se preterano (gotovo isključivo) propagira računar BBC B Za to treba imati razumevanja jer je seriju snimio upravo BBC pa je sasvim za očekivati da propagiraju mašinu koja nosi njihovo ime i donosi im određena profita (ili im je bar nekad donosila)

Computer Shop

Otvoren je i prvi Computer Shop u Jugoslaviji To je vest koja mora da obraduje i najveće pesimiste u računarskom pokretu U Ulici generala Ždanova 33 u Beogradu otvorena je radnja kojoj je jedina namena prodaja računarske opreme Ponuda je prilično raznovrsna za naše prilike i obuhvata računara („komodor", orao „lola" galaksija ivel ultra") dodatnu opremu (monitora kasetofone disk jedinice) rezervne delove i potrošni materijal programe (uglavnom obrazovni softver) kao i stručnu literaturu (osamdesetak knjiga iz oblasti kibernetike informatike i automatizacije)

Iako je Cõputer Shop tek nedavno počeo sa radom rezultati su očigledni radnja je po čitav dan puna posmatrača i kupaca koji prelistavaju literaturu razgovaraju sa prodavcem i naravno igraju igre na računarima koji su sve vreme uključeni S vremena na vreme kad gužva postane prevelika i novi radoznalci ne mogu da ulaze u radnju zato što je ona već prepuna prodavci gase računare i čekaju da se gužva bar malo raziđe

Uskoro Computer Shop treba da se preseli u nove veće prostorije kao i da proširi asortiman roda koji nije preterano veliki Takođe se pominje novi konkurs za prodavače koji bi trebao da pomogne da se tu zaposle ljudi koji zaista znaju nešto o računarima i koji budućim vlasnicima i korisnicima mogu da pruže pravu pomoć

Razvuci lestve i sačekaj raku

[illegible] časopisa Moj mikro objavljen je [illegible] nekoliko softverskih kaseta koje je u srpskohrvatskoj i slovenačkoj varijanti izdala Suzy iz Zagreba Uz stručne primedbe Moj mikro upućuje ovom poduhvatu i ozbiljne jezička zamerke Jezikovno razsodišče arbitraža koja se s vremena na vreme oglasi u Ljubljani verovatno će opet imati povod za raspravu jer u propratnoj knjižici ima sh-slovenačkih umotvorina Zaista šteta da se kad je već uložen toliki trud niko nije dosetio da nađe nekog Slovenca a ne „Slovenca" da stvar prevede valjano " I još i ovaj program je preveden na slovenački jezik pa i on obiluje jezičkim cvebama (napr ' planet lahko napadnuta tak kader je)

U istom broju „Mirka mogu se međutim naći i ovakve „sh-slovenačke" umotvorine „Tastatura je zaključena jedinica. , a matrika je ovde 6x10 tačaka matrike mora biti „Največa odlika novog Amstradovog računara je cena tak da na primer ' što bi izvesno smanjilo odstotak "; Pamet je prvo pritisnuti 0 i ENTER " dok duga piša ali crta padobran će popustiti in paščeš na zemlju " „Razvuci lestve i sačekaj raku

P S Nagradno pitanje br 1 Šta je cveba?



Amstrad

Kako raste bejzik

Amstrad/Schnaider CPC464 je bez sumnje jedna od mašina sa najmoćnijim setom bejzik instrukcija. Tu se odmah mogu postaviti dva pitanja: da li su nam potrebne još neke instrukcije, i ako jesu kako ih možemo pridodati postojećim.

Vidimo da se RAM i ROM preklapaju na dva mesta. Na adresama #0000 do #4000 se nalazi donji ROM (Lower ROM) u kome se nalaze sve rutine koje opslužuju hardver. Tu se ne nalazi bejzik. Gornji ROMovi (Upper ROMs) se preklapaju sa video memorijom (adrese #C000 do #10000). Ako vaš CPC464 nije proširivan u njemu se nalazi samo jedan gornji ROM sa bejzikom. Dalekovidi konstruktori su predvideli mogućnost dodavanja još 252 ROMa od po 16K koji se preklapaju sa gornjim ROMom. U njih se mogu smestiti FORTH, CP/M sistem, tekst procesor, asembler ili koji god program zaželite.

Pogledajmo prvo CPC464 memorijsku mapu			
ADRESA	RAM	ADRESA	ROM
#10000		#10000	
	VIDEO MEMORIJA		GORNJI ROMovi
#C000		#C000	
	STEK PARAM		
#A796	HIMEM		
#4000		#4000	
			DONJI ROM OS
#0000		#0000	

RSX (resident system extension) je vrlo sličan dodatnom ROMu. Razlika je u tome što se RSX nalazi u RAMu i kao takav mora da se učita svaki put po uključivanju računara, naravno ukoliko su nam potrebne rutine koje on poziva. Jednom učitan RSX treba inicijalizovati pozivom rutine KL LOG EXT =BCD1. Posle inicijalizacije operativni sistem zna da ukoliko bejzik ne prepozna naredbu koja se upravo obrađuje pogleda u tabelu naredbi RSXa.

Pogledajmo kako bi izgledala tabela naredbi RSXa koji bi proširio grafička mogućnosti ugrađenog bejzika.

Računar prepoznaje spoljašnju komandu po uspravnoj crti koja joj prethodi. Recimo komande iz gornjeg primera bi se u bejzik programu pisale kao |CIRCLE, |TRAINGLE itd. Vertikalna crta se dobija kao SHIFT/@.

	DW	NAME TABLE	adresa tabele imena komandi
	JP	DRAW CIRCLE	0
	JP	DRAW TRIANGLE	1
	JP	FILL AREA	2
		itd	
NAME TABLE	DB	CIRCL E +#80	0
	DB	TRIANGL E +#80	1
	DB	FIL L +#80	2
		itd	
	DB	0	kraj tabele imena komandi

Dobro, reći ćete, ali nekim komandama su potrebni parametri. Parametri se dodaju RSX komandi u obliku numeričke vrednosti, izračunate vrednosti nekog numeričkog izraza ili adrese. Broj i tip parametara mora biti usklađen između bejzik programa i RSX komande jer bejzik ne vrši provere. Svaki parametar je dužine mašinske reči (dva bajta) a interpretacija zavisi od tipa parametra.

Celobrojni izraz: komplement dvojke vrednosti integera.

Realni izraz: realni rezultat.

Promenljiva: adresa vrednosti promenljive (za niz adresa opisnog bloka niza). Opisni blok niza je tri bajta dugačak. Bajt 0 sadrži dužinu niza a sadržaj bajta 1 i 2 je početna adresa niza.

Pri skoku na rutinu DRAWCIRCLE na primer registar A sadrži broj parametara a IX adresu gde se oni nalaze. Ako imamo n parametara tada i-ti parametar ima pomeraj od (n-i)*2 od adrese u indeksnom registru, dakle prvi parametar ima najveći pomeraj, a poslednji parametar je na adresi koju pokazuje IX.

Kako sve to izgleda na konkretnom primeru? Listing 1 predstavlja RSX sa programom za popunjavanje konveksnih likova. Kako program radi: pošto komandom |FILL x,y prenesemo x i y koordinatu tačke koja se nalazi u nekoj konveksnoj konturi program ispituje sve tačke levo od tačke x,y dok ne dođe do tačke koja pripada konturi i čija koordinata zapamti. Zatim analogno ispituje sve tačke desno i kada dođe do krajnje desne tačke povlači liniju između graničnih tačaka. Zatim postavlja x koordinatu nove početne tačke na sredinu rastojanja granične leve i desne tačke, smanjuje y koordinatu i ispituje da li se nova početna tačka nalazi u konturi. Ako se ne nalazi u konturi ceo postupak počinje iz početka. Ako se nalazi vraća se u program koji ga je pozvao.

Kako uneti program? Ako imate asembler (na primer DEVPAC) možete direktno uneti listing. Nemojte ga odmah prevoditi već ga sa P10,1030 ime sačuvajte na traku. Zatim ga asemblirajte i sa O, ime opet sačuvajte na traku. Prilikom korišćenja bilo bi dobro da prvo spustite HIMEM, učitate pro-

```
10 GRA TE:  EQU   #BBF0
20 GRA_LI:  EQU   #BBF6
30 GRA_SE:  EQU   #BBD1
40          ORG   42723             ;MEMORY HIMEM-100
50          LD    BC,RSX
60          LD    HL,KER
70          PUSH  DE
80          CALL  #BCD1             ;INICIJALIZACIJA RSX
90          POP   DE
100         RET                     ;KRAJ INICIJALIZACIJE
110 RSX:    DEFW  NAME_TABLE        ;ADRESA TABELE NAREDBI
120         JP    FILL
130 NAME T: DEFM  "FIL"             ;POCETAK TABELE NAREDBI RSX
140         DEFB  "L"+#80           ;ZADNJE SLOVO JE SHIFTOVANO ZA #80
150         DEFB  0                 ;KRAJ TABELE NAREDBI
160 FILL:   PUSH  AF                ;POCETAK FILL RUTINE
170         PUSH  BC                ;SACUVAJ REGISTRE
180         PUSH  DE
190         PUSH  HL
200         PUSH  IX                ;IX U HL
210         POP   HL
220         LD    E,(HL)            ;UZMI PARAMETRE OD BASICA
230         INC   HL
240         LD    D,(HL)
250         LD    (POCY),DE
260         INC   HL
270         LD    E,(HL)
280         INC   HL
290         LD    D,(HL)
300         LD    (POCX),DE
310         CALL  GRA_GE            ;NADJI INK
320         LD    (INK),A
330         LD    DE,(POCX)         ;POSTAVI KOORDINATE
340         LD    HL,(POCY)
350 ALFA:   PUSH  DE                ;TEST TACKE
360         PUSH  HL
370         CALL  GRA_TE
380         POP   HL
390         POP   DE
400         CP    0
410         JR    NZ,BETA
420         DEC   DE                ;IDI LEVO
430         JR    ALFA
440 BETA:   LD    (LASTX),DE        ;SACUVAJ KOORDINATE
450         LD    (LASTY),HL        ;KRAJNJE LEVE TACKE
460         LD    DE,(POCX)
470         LD    HL,(POCY)
480 GAMA:   PUSH  DE                ;TEST TACKE
490         PUSH  HL
500         CALL  GRA TE
510         POP   HL
520         POP   DE
530         CP    0
540         JR    NZ,DELTA
550         INC   DE                ;IDI DESNO
560         JR    GAMA
570 DELTA:  PUSH  DE                ;NACRTAJ LINIJU
580         PUSH  HL
590         LD    DE,(LASTX)
600         LD    HL,(LASTY)
610         CALL  GRA TE
620         POP   HL
630         POP   DE
640         PUSH  DE
650         PUSH  HL
660         LD    A,(INK)
670         CALL  GRA LI
680         POP   HL
690         POP   DE
700         EX    DE,HL             ;DE =LASTX+(DE-LASTX)/2
710         LD    BC,(LASTX)
720         SBC   HL,BC
730         LD    A,H
740         SRL   A
750         LD    H,A
760         LD    A,L
770         RRA   A
780         LD    L,A
790         ADD   HL,BC
800         EX    DE,HL
810         DEC   HL                ;HL=HL-2
820         DEC   HL
830         LD    (POCX),DE
840         LD    (POCY),HL
850         PUSH  DE
860         PUSH  HL
```

```
870          CALL GRA_T0
880          POP  HL
890          POP  DE
900          CP   0
910          JR   NZ,KRAJ
920          JR   ALFA
930 KRAJ:    POP  HL
940          POP  DE
950          POP  BC
960          POP  AF
970          RET
980 POCX:    DEFW 0
990 POCY:    DEFW 0
1000 INK:    DEFB 0
1010 LASTH:  DEFW 0
1020 LASTV:  DEFW 0
1030 STR:    DEFM "AAAA"
```

```
10 REM SMESTANJE MASINSKOG DELA
20 MEMORY 42722
30 FOR I=42723 TO 42723+187
40 READ K$
50 K=VAL("&"+K$)
60 A=VAL(STR$(K))
70 POKE I,A
80 NEXT
90 CALL 42723
100 DATA 01,EF,A6,21,93,A7,D5,CD,D1,BC,D1,C9
110 DATA F4,A6,C3,F9,A6,46,49,4C,CC,00,F5,C5
120 DATA D5,E5,DD,E5,E1,5E,23,56,ED,53,8C,A7
130 DATA 23,5E,23,56,ED,53,8A,A7,CD,E1,BB,32
140 DATA 8E,A7,ED,5B,8A,A7,2A,8C,A7,D5,E5,CD
150 DATA F0,BB,E1,D1,FE,00,20,03,1B,18,F2,ED
160 DATA 53,8F,A7,22,91,A7,ED,5B,8A,A7,2A,8C
170 DATA A7,D5,E5,CD,F0,BB,E1,D1,FE,00,20,03
180 DATA 13,18,F2,D5,E5,ED,5B,8F,A7,2A,91,A7
190 DATA CD,F0,BB,E1,D1,D5,E5,3A,8E,A7,CD,F6
200 DATA BB,E1,D1,EB,ED,4B,8F,A7,ED,42,7C,CB
210 DATA 3F,87,7D,CB,1F,4F,09,EB,2B,2B,ED,53
220 DATA 8A,A7,22,8C,A7,D5,E5,CD,F0,BB,E1,D1
230 DATA FE,00,20,02,18,97,E1,D1,C1,F1,C9,00
```

```
10 REM DEMO PROGRAM
20 CLS
30 MOVE 0,100
40 DRAW 100,100
50 DRAW 50,200
60 DRAW 0,100
70 |FILL,50,199
80 FOR I=0 TO 2*PI STEP 0.01
90 PLOT 320+40*COS(I),200+90*SIN(I)
100 NEXT
110 |FILL,320,200:REM POLA ELIPSE
120 |FILL,320,289:REM I DRUGA POLOVINA
```

gram (naravno, mašinsku verziju) izvršite inicijalizaciju (sa CALL 42723) i zatim učitavajte ili unosite bejzik programe koji koriste ovu rutinu. Za kasnije korišćenje je moguće sve ove korake (od spuštanja HIMEMa) automatizovati tako da se mašinski program i bejzik program koji ga koristi učitavaju zajedno sa trake. Ako nemate asembler unesite program sa hex-dampom. Po njegovom izvršenju u vašem računaru će biti mašinska verzija programa i to inicijalizovana. Bejzik programa brišite sa NEW ukoliko želite da RSX ostane u memoriji.

Ovde je objašnjeno i praktično pokazano kako se može proširiti set naredbi vašeg računara, i to bez beskrajnih USR, CALL i POKE naredbi. U RSX se mogu smestiti programi kao HARD-COPY((program za preslikavanje kompletnog sadržaja ekrana na štampač) ili rutine za opsluživanje modema, unutrašnji časovnik, kao i mnoge druge.

Srđan Stakić

SPEKTRUM

Zvuk iz „spektruma“

Postoji već nekoliko komercijalnih programa za snimanje i reprodukciju zvuka sa „spectrumom“, ali svima je zajednička mana nedostatak indikacije zvuka prilikom snimanja. U rutini čiji je listing dat indikacija je rešena slično kao kod komande LOAD sa linijama po borderu čime je dosta olakšano podešavanje optimalne jačine zvuka.

Pošto „spektrum“ uma da razlikuje samo dva stanja na EAR ulazu (otuda naziv „Binarvoice“), važno je da zvuk koji se snima ima što manju dinamiku — najbolje je zvuk prvo snimiti na kasetofon sa automatskim regulisanjem nivoa snimanja, pa odatle vršiti snimanje u računar.

Rutina može biti smeštena bilo gde u gornjih 32K RAMa (u listingu treba dodati odgovarajući ORG), dok područje za upis zvuka u memoriju (labela BEGIN) može počinjati i u donjih 16K. Testiranje kraja područja se obavlja radi jednostavnosti u koracima od 256 bajta (to je 1/4 sekunde pri datom T). Za to služi labela ENDHB (bajt veće vrednosti).

```
10 ;
20 ;           BINARVOICE
30 ;
40 ; SNIMANJE
50      DI
60      LD HL,BEGIN      ;Početak područja
70      LD DE,SETX+1     ;upisa.
80      LD C,#FE         ;ULA port.
90 RDVTL LD (HL),0
100     LD B,#C6         ;Opkod za SET 0.
110 RDITL LD (DE),B
120     IN A,(C)
130     BIT 6,A
140     JR NZ,NOG
150 SETX SET 0,(HL)      ;Upis bita u memoriju
160     LD B,7
170     DJNZ $           ;Pauza
180     LD B,#0A
190 SIGN OUT (C),B       ;Border efekt
200     ADD B,8          ;Priprema za sledeći
210     CP 6             ;bit
220     JR NZ,RDITDE
230     INC HL           ;Sledeći bajt
240     LD B,ENDHB       ;Test kraja u kora-
250     CP H             ;cima od 256 bajta
260     JR NZ,RDVTL
270     EI
280     RET              ;Povratak u Basic
290 NOG LD B,T
300     DJNZ $
310     LD B,#F
320     JR SIGN
330 RDITDE LD B,2        ;Dodatno kašnjenje
340     DJNZ $           ;kod bitova unutar
350     JR RDITL         ;jednog bajta.
360
370 ; REPRODUKCIJA
380     DI
390     LD HL,BEGIN
400     LD DE,BITX+1
410     LD C,#FE
420 PDVTL LD B,#46       ;Opkod za BIT 0 .
430 PDITL LD (DE),B
440 BITX BIT 0,(HL)
450     JR Z,SOH
460     LD B,T+3
470     DJNZ $           ;Pauza.
480     LD B,#12
490 VOICE OUT (C),B
500     ADD B,8          ;Sledeći bit.
510     CP #86
520     JR NZ,7DIDEL
530     INC HL
540     LD B,ENDHB
550     CP H
560     JR NZ,PDVTL
570     EI
580     RET
590 SOH LD B,T+1
600     DJNZ $
610     LD B,7
620     JR VOICE
630 7DIDEL LD B,1
640     DJNZ $
650     JR PDITL
660
670 T EQU #10            ;Preporučena vremenska
680                      ;konstanta
```

Sa zadatom vremenskom konstantom T (24) potrošnja memorije je oko 1 K za sekundu zvuka. Ako T smanjimo, dobićemo nešto kvalitetniju reprodukciju ali čuda ne treba očekivati.

Ko želi može dodati program pomoću koga se na jednostavan način mogu menjati početak i kraj područja upisa vremenska konstanta itd.

Petar Putnik

Komodor 64

Disasembler

Svi disasembleri za komodor su izvedeni u okviru neke šire celine (monitori, HELP 64+) i ne pružaju poseban komfor u radu. To se posebno odnosi na generisani listing koji nema ASCII prezentaciju memorijskog sadržaja pa pri nailasku na neku tablicu u programu treba koristiti druge funkcije da bi se utvrdilo o čemu je reč. Disasembliranje većih programa zahteva stalno menjanje adresa ili ispis na štampač što često zahteva velike količine papira.

Ovaj program otklanja najveći broj navedenih nedostataka. Disasemblira zadati broj instrukcija (do 200) i zatim generiše spis po kome se može šetati. Kada se završi analiza deo se ako je to potrebno štampa i zatim disasemblira sledeća grupa instrukcija.

Pozivanje disasemblera se ostvaruje naredbom

SYS 36000 POČETNA ADRESA BROJ INSTRUKCIJA [STVARNA ADRESA KODA]

Stvarna adresa koda je opciona i omogućava relocirano disasembliranje. Kod se uzima sa te adrese a disasemblira kao da je na POČETNOJ ADRESI.

KONTROLNE SUME:

S 0 = 4644	S 25 = 3765
S 1 = 3883	S 26 = 5443
S 2 = 5552	S 27 = 5387
S 3 = 4378	S 28 = 5473
S 4 = 4855	S 29 = 2587
S 5 = 4533	S 30 = 5331
S 6 = 4255	S 31 = 1627
S 7 = 4188	S 32 = 1458
S 8 = 4887	S 33 = 1874
S 9 = 3888	S 34 = 1151
S 10 = 4868	S 35 = 1187
S 11 = 5364	S 36 = 1128
S 12 = 3415	S 37 = 741
S 13 = 4525	S 38 = 782
S 14 = 4163	S 39 = 508
S 15 = 3686	S 40 = 558
S 16 = 4688	S 41 = 280
S 17 = 4805	S 42 = 336
S 18 = 3761	S 43 = 388
S 19 = 5778	S 44 = 334
S 20 = 4418	S 45 = 384
S 21 = 2874	S 46 = 364
S 22 = 3346	S 47 = 318
S 23 = 4058	S 48 = 318
S 24 = 3165	S 49 = 838
	S 50 = 1588

Pri pojavi listinga na ekranu tasterima KURSOR DOLE/GORE listing se pomera za jednu instrukciju dok razmaknica i SHIFT+razmaknica vrše pomeranje za ceo ekran. Grupe instrukcija se štampa pritiskom na F7 i to u dve kolone tako da na jednu stranu u proseku stane preko 300 bajtova disasembliranog programa. F8 vraća kontrolu bejziku.

Ispis svih vrednosti je u decimalnom obliku sa sledećim formatom:
adresa instrukcije
niži bajt adresa
ascii prezentacija koda instrukcije
kod instrukcije
mnemonik

Adresa će povremeno biti preklopljena sa višim bajtom.

Zoran Životić

```
1 REM'——— D I S A S E M B L E R ———
2 REM
3 REM          C64
4 REM
5 REM   SYS36000,ADR,00,INST [,ORB]
6 REM
7 REM AUTORI ZORAN ZIVOTIC (85)
8 REM'————————————————————————————
10 REM <<<<< PROVERA DATA LINIJA >>>>>
11 RESTORE:FORI=100TO303STEP4:S=0
12 FORJ=1TO32:READB:S=S+B:NEXTJ
13 PRINT"S'1/4-25'+'5;
14 A$='D':INPUT' OK';A$:IFA$='D'GOTO17
15 PRINT' LIST'I'-'I+3:POKE631,19
16 POKE632,13:POKE198,2:SYS42115
17 NEXTI:PRINT"I,OK'
18 RESTORE:S=0:FORF=1TO1040:READB:S=S+B
19 NEXTF:IFS=135333THENPRINT'2,OK':END
20 PRINT'2 GRESKA!!':END
21 REM
30 REM <<<<< KREIRANJE PROGRAMA >>>>>
31 CLR:POKE52,140:POKE56,140:CLR:RESTORE
32 FORF=36000TO37630:READB:POKEF,B:NEXT
33 NEW
100 DATA 32,253,174, 32,138,173, 32,247
101 DATA163,165, 20,133,245,133,251,165
102 DATA 21,133,250,139,252, 32,253,174
103 DATA 32,150,165,138, 41,254,208,  2
104 DATA165,  2,201,201,144,  2,165,200
105 DATA133, 80,100,  0,177,122,201, 44
106 DATA208, 17, 32,115,  0, 32,138,173
107 DATA 32,247,163,165, 20,133,249,165
108 DATA 21,133,250,169,  0,138,253,168
109 DATA100,133,254,160,  0,188, 39,145
110 DATA253,200,200 248,230,254,165,254
111 DATA201,162,208,241,165,160,133,254
112 DATA169,192,168,146, 32, 38,171,165
113 DATA 88,133, 70, 32, 84,142,165,253
114 DATA 24,105, 40,133,253,165,  0,101
115 DATA254,133,254,150, 70,208,208,160
116 DATA  0,133,253,133, 70,133,  2,165
117 DATA160,133,254, 32,187,141, 32,228
118 DATA255,240,251,133,  2,201, 17,208
119 DATA 27,165, 70, 24,105, 24,157, 69
120 DATA176,230,165,253 24,105, 40,133
121 DATA253 105   0,101 254,133,254,230
122 DATA 70, 76, 43,141 201,145,208, 25
123 DATA100, 70,202,224,255,240,207,134
124 DATA 70,165,253, 56,233, 40,133,253
125 DATA165,254,233,  0,133,254, 76, 43
126 DATA141,201, 32,208, 29,165, 70, 24
127 DATA185, 47,157, 89,170,176,233, 22
128 DATA133, 70,165,253, 24,105,130,133
129 DATA253,165,254,105,  3,133,254, 70
130 DATA 43,141,201,160,200, 25,165, 70
131 DATA 56,233, 24,144,145,133, 70, 56
132 DATA165,253,233,162,133,253,165,204
133 DATA233,  3,133,254, 70, 43,141,201
134 DATA133,209,  3, 76,207,140,201,136
135 DATA200   3, 76, 92 143,201,140,200
136 DATA  1, 80, 70, 40,141,165,  0,133
137 DATA160,165,  1, 41,254,133,  1,160
138 DATA 10, 32,210,255,165,253, 72,165
139 DATA254, 72,169,  4,133, 34,169,162
140 DATA133, 35,160,  0,177,253, 32,210
141 DATA255, 30,  5,142,160, 35,208,240
142 DATA130, 34,200 230,164,133,254,164
143 DATA133,203,165,  1,  0,  1,133,  1
144 DATA169,  0,133,212, 36,230,253,200
145 DATA  2,230 254  30,230,249 200,  2
146 DATA230,250,208,251,200,245,236,202
147 DATA 30,134, 80,133, 80,132,  2,162
148 DATA144, 58, 32, 73,169, 32,221,160
149 DATA182,  1,165,  1,  1,240,  0,232
150 DATA 36,  2, 10,240,230   2, 40,242
151 DATA160,127  37,  2,133,  2,168,160
152 DATA  0   1,145,253 136,202,200,247
153 DATA 30,160,  0 177,240,133 151  41
154 DATA 56,200,  4,160, 32,133,151,130
155 DATA160,165,151,145,253, 90,165,252
156 DATA160,251,160,  4, 32, 25,142,165
157 DATA251, 41, 15,209, 14,166,252,160
158 DATA120, 32, 25,142,104,  2,208,165
159 DATA 45,145,253,160,  0,165,251,160
160 DATA  0, 32, 25,142,160,  0,177,245
161 DATA133,150,160, 16,170,165,  0,160
162 DATA 10, 32, 25,142,165,150 170,160
163 DATA152,144,170,160, 20,169,228,143
164 DATA140,253,232,206,162, 29,144,245
165 DATA166,150,169,162,145,208,  7,169
166 DATA  0,133,150, 76, 57,143,133,151
167 DATA176 202,160, 30,169,137,144,145
168 DATA253,206,232,160,137,144,133,150
169 DATA160,  1,177,240, 72,170,169,  0
170 DATA160, 20, 32, 25,142,169,  0, 72
171 DATA165 150 201, 49,208, 14,160,  2
172 DATA164,177,249, 72,170,165,  0,160
173 DATA 24, 32, 25,142,165,150,201, 50
174 DATA240, 13,164,169,164,170,132,160
175 DATA150, 32, 25,142, 70, 30,143,165
176 DATA251, 24,165,  2,133, 30,165,252
177 DATA169,  0,133, 35,164,133, 41,164
178 DATA133, 40, 36, 40, 10,  2,160, 41
179 DATA 24,165, 30,101, 40, 72,165, 38
180 DATA101, 41, 72, 70,242,142,166,151
181 DATA164,  2,200,232,160,137,144,145
182 DATA253,130, 50,220,151,201,  3,208
183 DATA241,162, 10, 32, 73,142,165,150
184 DATA240, 22, 32, 12,140,162, 11, 32
185 DATA 73,142,165,150,201, 40,208,  8
186 DATA 32, 12,142,162, 12, 32, 73,142
187 DATA 32, 12,142, 30,165,  0,133, 34
188 DATA133, 30,160,160,133, 35,133, 37
189 DATA165, 80, 74,133, 70,170,165, 36
190 DATA 24,165, 40,133, 36,165, 37,165
191 DATA  0,133, 37,202,200,240,165,  1
192 DATA 41,254,133,  1,165,  4, 32,177
193 DATA255,168, 90, 32,147,255,160,  0
194 DATA177, 34, 32,160,255,200,130, 40
195 DATA200,240,160,  0,177, 30, 32,160
196 DATA255,200,152, 40,200,240,165, 34
197 DATA 24,165  40,133, 34,165, 35,165
198 DATA  0,133, 35,165, 30, 24,165, 40
199 DATA133, 30,165, 37,165,  0,133, 37
200 DATA150, 70,200,202,162,  0,165, 13
201 DATA 32,160,255,202,200,240, 32,174
202 DATA255,169,  1,  5,  1,133,  1,165
203 DATA  0,133,160, 76, 40,141, 45, 45
204 DATA 45, 70, 80, 85, 76, 80, 80, 70
205 DATA 80, 85, 83, 84, 85, 83, 84, 80
206 DATA 83, 84, 69, 87, 77, 68, 87, 80
207 DATA 88, 87, 80, 80, 73, 78, 87, 80
208 DATA 69, 87, 65, 80, 67, 83, 80, 87
209 DATA 65, 78, 80  78, 82, 65, 68, 75
210 DATA 82, 85, 83, 70, 70, 83, 82, 82
211 DATA 75, 76, 82, 78, 82, 80, 73, 84
212 DATA 74, 83, 82, 74, 77, 80, 82, 84
213 DATA 83, 82, 84, 73, 79, 79, 80, 80
214 DATA 82, 75, 80, 76, 68, 80, 76, 65
215 DATA 80, 72, 80, 80, 72, 65, 84, 85
216 DATA 80, 84, 80, 85, 84, 85, 80, 84
217 DATA 85, 85, 84, 83, 80, 84, 80, 83
218 DATA 68, 80, 68, 80, 89, 65, 73, 78
219 DATA 80, 73, 78, 80  66, 67, 87, 80
220 DATA 67  83, 80, 78, 80, 80, 80, 81
221 DATA 80, 80, 76, 89, 77, 73, 68, 85
222 DATA 87, 80, 80, 83, 87, 76, 67, 83
223 DATA 69, 87, 67, 76, 68, 83, 80, 80
224 DATA 87, 78, 73, 83, 89, 73, 67, 76
225 DATA 80, 35, 40, 32, 32, 30, 32, 40
226 DATA 32, 32, 32, 30, 40, 44, 80, 32
227 DATA 32, 40, 44, 85  32, 32, 48, 32
228 DATA 32, 32, 32, 45, 44, 88, 32, 32
229 DATA 45, 44, 89, 32, 40, 40, 44, 80
230 DATA 41, 40, 40, 41, 44, 69, 40, 40
231 DATA 41, 32, 32, 32, 50, 32, 32, 30
232 DATA 81, 45,  0,  0,  0, 45, 51,  0
233 DATA 88, 45, 51,  0,  0, 45, 51,  0
234 DATA130, 45,  0,  0   0, 45, 51,  0
235 DATA150, 45,  0,  0,  0, 45, 51,  0
236 DATA 69, 42,  0,  0, 63, 42, 57,  0
237 DATA 84, 42, 57,  0, 63, 42, 57,  0
238 DATA141, 42,  0,  0,  0, 42, 57,  0
239 DATA153, 42,  0,  0,  0, 42, 57,  0
240 DATA 75, 40,  0,  0,  0, 40, 54,  0
241 DATA 83, 40, 54,  0, 88, 40, 54,  0
242 DATA144, 40,  0,  0,  0, 40, 54,  0
243 DATA160, 40,  0,  0,  0, 40, 54,  0
244 DATA 72, 36,  0,  0,  0, 36, 80,  0
245 DATA 87, 36, 80,  0, 80, 36, 60,  0
246 DATA147, 36,  0   0   0, 36  80,  0
247 DATA165, 36,  0,  0,  0, 36, 80,  0
248 DATA  0, 12,  0,  0, 10, 12, 15,  0
249 DATA117,  0, 33,  0, 10, 12, 15,  0
250 DATA120, 12,  0,  0, 10, 12, 15,  0
251 DATA185, 12,111,  0,  0, 12,  0,  0
252 DATA  9,  3   6,  0,  9,  3,  0,  0
253 DATA102,  3, 30,  0,  0,  3,  0,  0
254 DATA129,  3,  0,  0,  0,  3,  0,  0
255 DATA169,  3,100,  0,  0,  3,  0,  0
256 DATA 27, 21,  0,  0, 27, 21, 33,  0
257 DATA123  21,114,  0, 27, 21, 33,  0
258 DATA132, 21,  0,  0,  0, 21, 33,  0
259 DATA100  21,  0,  0,  0, 21, 33,  0
260 DATA 24, 30,  0,  0, 24, 30, 30,  0
261 DATA120, 39, 70,  0, 24, 30, 30,  0
262 DATA135, 39,  0,  0,  0, 39, 30,  0
263 DATA159, 35,  0,  0,  0, 30, 30,  0
264 DATA  0, 36,  0,  0,  0,  6,  6,  0
265 DATA  0,  1,  0,  0,  0, 21, 21,  0
266 DATA 51, 41,  0,  0,  0, 11, 11,  0
267 DATA  0, 31,  0,  0,  0, 26, 28,  0
268 DATA 21, 36,  0,  0,  0,  0,  6,  0
269 DATA  0,  1,  0,  0, 21, 01, 21,  0
270 DATA 51, 41,  0,  0,  0, 11, 11,  0
271 DATA  0, 31,  0,  0,  0, 20, 60,  0
272 DATA  0, 36,  0,  0,  0,  0,  0,  0
273 DATA  0,  1,  0,  0, 21, 21, 21,  0
274 DATA 51, 41,  0,  0,  0, 11, 11,  0
275 DATA  0, 31,  0,  0,  0, 20, 26,  0
276 DATA  0, 30,  0,  0,  0,  0,  0,  0
277 DATA  0,  1,  0,  0, 40, 21, 21,  0
278 DATA 51, 41,  0,  0,  0, 11, 11,  0
279 DATA  0, 31,  0,  0,  0, 20, 20,  0
280 DATA  0, 30,  0,  0,  6,  8,  0,  0
281 DATA  0,  0,  0,  0, 21, 21, 21,  0
282 DATA 51, 41,  0,  0, 11, 11, 10,  0
283 DATA  0, 31,  0,  0,  0, 20,  0,  0
284 DATA  1, 30,  1,  0,  0,  6,  0,  0
285 DATA  0,  1,  0,  0, 21, 21, 21,  0
286 DATA 51, 41,  0,  0, 11, 11  18,  0
287 DATA  0, 31,  0,  0, 26, 26, 31,  0
288 DATA  1, 36,  0,  0,  0,  0,  0,  0
289 DATA  0,  1,  0,  0, 21, 21, 21,  0
290 DATA 51, 41,  0,  0,  0, 11, 11,  0
291 DATA  0, 31,  0,  0,  0, 30, 20,  0
292 DATA  1, 30,  0,  0,  0,  0,  6,  0
293 DATA  0,  1,  0,  0, 21, 21, 21,  0
294 DATA 51, 41,  0,  0,  0, 11, 11,  0
295 DATA  0, 31,  0,  0,  0, 26, 20,  0
296 DATA 17, 17, 17, 17, 17, 17, 17, 17
297 DATA 17, 17, 17, 17, 17, 17, 17, 17
298 DATA 17, 17, 17, 17, 17, 17, 17, 17
299 DATA 17, 32, 10, 87, 80, 83, 82, 47
300 DATA 83, 80, 65, 87, 80, 47, 70, 40
301 DATA 47, 70, 55, 47, 70, 56, 32, 32
302 DATA 32, 32, 32, 32, 32, 80, 47, 50
303 DATA 32, 32, 32, 32, 32, 32, 32, 32
304 DATA 32,146,140,  0,  0,  0,  0,  0
READY.
```

Komodor 64

Datoteka novinskih članaka

Ovaj program omogucava stvaranje biblioteka podataka pomoću koje se lako može naći u kom smo časopisu novinama ili knjizi pročitali nešto što je interesantno ili je iz oblasti naše profesije ili hobija.

Podaci se mogu formirati i pozivati prema nazivu časopisa, temi, oblasti interesovanja, naslovu članka ili autoru a moguće je i njihova kombinacija. Podaci se mogu lapisati na printeru.

Posle ukucavanja potrebno je počev od linija 2000 uneti željena

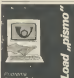


[illegible]

podatka (primer su nekoliko DATA linija na kraju programa) sn m i ih na kasetu ili disketu i ovu opetaciju ponovili uvek kada se doda neki nov podatak

Program se startuje sa RUN a MENI govori za sebe Jedino opcije I i ILI traži objašnjenje I znači da je moguće tražiti vezano dva podatka a unos se sa 1 2 (ili druge potrebne mogućnosti) ILI znači izbor od dve željane mogucnosti a takođe se unosi sa 1 2

Odgovon na pitanje manija mogu se unositi skraćano ali BEZ upotrebe zvezdice (*) Ukoliko se želi svi DATA podaci dobijaju se samo sa RETURN posle izbora iz menija



Priorema
Branko Đaković

Na otpad i sa otpada

Ono što je Dragan uradio sad u „osmici" nije nimalo lepo Radi se o njegovom tekstu u rubrici Šta ima novo pod naslovom Računari na otpadu Dejan se tu lepo zabavlja i provaljuje a zapravo upropašćuje šanse da još neke naše firme na taj način dobiju računara Možda taj način nije baš 150% legalan ali ipak jeste nekakav način da naša tehnološki osiromašena privreda malo poboljša svoj status, informacijski gledano

Moja firma je bila na samom pragu da nabavi jedan računar tehnikom opisanom u članku Sad više nisam baš siguran u tako nešto Ne postoji ništa nelegalno u kupovini računara sa otpada lako to može malo da miriše na mućku' Voleo bih da znam da li su baš svi članovi redakcije Računara došli do svojih kompjutera baš 100% legalnim putem

D D Beograd

Dejan ne kritikuje nego samo predlaže bolje i — jeftinije rešenje Što se tiče zakonodavca, budite bez brige — niko vas neće sprečavati ni da svoj računar bacite na otpad niti da ga kupite u njemu

Hakeri u namilosti

Ima nesto jako cudno u vezi sa vašim intervjuima Ako izuzmemo onaj sa Stivenom Džobsom koji i nije bio baš vaš intervju, u svim ostalim razgovorima postoji jedan zajednički element da ne kažem potprogram U svakom od njih se lamava ponižava i nipodaštava „mali" programer tj haker Svako ima pravo da ima svoje mišljenje o hakerima ali nipošto nije u redu da ton tog mišljenja bude ispod nivoa propisanog kulturom jednog javnog glasila Ja lično nisam haker, ali moram priznati da mi takve stvari smetaju Spuštanje hakerima bi moglo da se obavi mnogo elegantnije lepše pristojnije i ako ćemo iskreno mnogo duhovitije nego što to čine vaši eminentni gosti Ako sam ja pogrešno protumačio ton ljudi koje ste intervjuisali izvinite Bilo bi prijatno videti bar nekog visokog stručnjaka KAKO GOVORI O HAKERIMA SA TOPLINOM I RAZUMEVANJEM

Srećko Milićević
Otona Župančiča 29
Novi Beograd

Put u središte ROM-a (5)

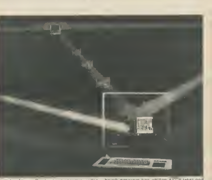
u svetu editora

U „lične stvari bejzika" o kojima smo govorili u pretprošlom broju „Računara" spada i pisanje editora koji treba da omogući brzo i jednostavno unošenje i ispravljanje budućih programa. Zato ćemo u ovom nastavku našeg putovanja u središte ROM-a posvetiti pažnju konstrukciji editora raznih popularnih kompjutera i razmotriti strukturu interpretera, odnosno kompajlera, pripremajući se za ozbiljan posao koji nas očekuje — pisanje interpretera za jednostavni kompjuterski jezik LOGO

Editorom nazivamo bilo koji program za obradu teksta, pri čemu taj tekst u specijalnom slučaju može da bude program pisan na bejziku ili nekom drugom programskom jeziku. Ukoliko opremamo računar koji će „razumeti" nekoliko programskih jezika, brzo ćemo se zapitati da li je pametno konstruisati poseban editor za svaki od njih. Jedan od argumenata za je činjenica da svaki jezik ima neke svoje karakteristike tako da će različite opcije editora dobro poslužiti. Dva glavna argumenta 'protiv' su da će svaki od editora zauzimati neki memorijski prostor u ROM-u i da će, što je možda još neprijatnije, siroti korisnik našeg računara morati posebno da se navikava na rad svakoga od njih. Znajući da je vrlo teško raditi danas sa jednim a sutra sa drugim editorom, morali bismo da se odlučimo za pisanje sličnih editora, a zatim da postavimo pitanje zašto da ne napišemo univerzalni editor velikih mogućnosti koji će zauzimati jedan od paralelnih jezičkih ROM-ova? Obzirom da će veličina ovog editora biti praktično jednaka veličini bejzik interpretatora, on će moći da ponudi mnogo opcija koje će zadovoljavati potrebe raznih programskih jezika. Korisnik će tada prvo startovati editor, zatim unositi i ispravljati program, a potom aktivirati interpretator (ili kompajler) nekog programskog jezika i tako ga izvršiti. Kao dodatni bonus korisnik će moći da koristi ovaj editor za obično pisanje i tako biti u prilici da uštedi nešto novca. Dokaz da je ovakav sled razmišljanja logički ispravan možemo da nađemo u činjenici da je primenjen kod svih (bez izuzetka) velikih kompjuterskih sistema. Dokaz da je pogrešan možemo da nađemo u činjenici da nije primenjen praktično ni kod jednog kućnog računara. U čemu je problem?

Jedan za sve

Glavni razlog za odustajanje od koncepcije „jedan editor za sve jezike" kod kućnih računara je sasvim ljudske prirode: kompjuteri se opremaju bejzik interpretatorom i time se završava njihov razvoj što omogućava brz izlazak na tržište. Svaki od konstruktora računara se, jasno, nada da će njegova mašina postići veliku popularnost i da će za nju biti pisano mnogo programa, ali razonuja „o tom potom", kada bude došlo vreme programi će se već napisati a možda će taj problem rešiti nezavisne sof-



tverske firme. Da bi se pripremio jedan editor za razne programske jezike treba zamisliti veoma fleksibilnu organizaciju memorije, obezbediti razna radna prostora dovoljnih ali ne i prevelikih dimenzija i što je najvažnije fiksirati memorijsku adresu od koje će se smeštati tekstovi programa, svi budući kompajleri i interpretatori će ga tražiti počevši od te adrese (kao alternativu možemo da uvedemo neku sistemsku promenljivu koja će se nalaziti na fiksnom mestu). Vidimo da ovakvu organizaciju nije lako zamisliti unapred tako da bude dovoljno efektna. Pa ipak, može se pribeći malim trikovima.

Da bi obezbedili međusobnu kompatibilnost raznoraznog softvera konstruktori BBC-ja su pripremili naredbe operativnog sistema SPOOL i EXEC. Ako na primer otkucate SPOOL PROBA na kaseti ili disku će biti otvorena datoteka PROBA u koju će biti upisivane sve vaše buduće akcije: računar jednostavno pamti sve tastere koje ste pritisnuli. Kada docnije budete upotrebili EXEC PROBA sve te akcije će biti ponovljena kao da ste čitavu seansu još jednom izvali sa tastature. Kako ovo može da se iskoristi? Aktivirajte na primer tekst procesor i u njemu pripremite i na disk snimite bejzik program kao običan ASCII tekst pod imenom PROG. Onda se vratite u bejzik i sa EXEC PROG posmatrajte mali crtani film: računar će se ponašati kao da sa tastatura kucate čitav program. Za docnije ćete ispravke ponovo startovati tekst procesor, učitati PROG sa diskete i, pošto ste uradili sve što ste želeli, ponovite prebacivanje programa u nadležnost bejzik interpretatora. U teoriji ovaj trik radi sasvim lepo, u praksi se pojavljuju problemi. Da bi naime, SPOOL/EXEC kombinacije bile efektne, brzina prenosa podataka između operativne i spoljne memorije mora da bude veoma velika, jasno je da je brzina kasetofona nedovoljna a i disk se pri čestim ispravkama programa pokazuje dosadno sporim. Zato ovakvo rešenje ima upotrebnu vrednost tek kod većih sistema koji su opremljeni hard diskovima.

Kada bismo uspeli (a to nikako nije nemoguće) da naš računar izvorne tekstove programa koji su pisani na bilo kom jeziku pripreme kao ASCII fajlova i smešta ih od neke fiksne memorijske lokacije, primena jednog editora za sve jezike bi i dalje bitno umanjila efikasnost našeg sistema. U prošlom broju „Računara" smo videli da se bejzik program, radi bržeg izvršavanja, to-

kanizuje tako što se naredbe zamenjuju kodovima s brojevi binarnom reprezentacijom. Besmisleno je očekivati da će svi jezici na našem računaru imati iste naredbe koje će se na isti način kodirati, pa bi upotreba univerzalnog editora najpre smanjila dimenzija programa koji može da se izvršava (ASCII tekst programa zauzima daleko više prostora od tokenizivane verzije), a zatim usporila njegovo izvršavanje. Usporavanje bi moglo da se eliminiše tako što bi se tokenizacija izvršila kada otkucamo RUN ali je to već neki vid prevođenja (kompilacije) programa. Veliki kompjuterski sistemi rade uglavnom sa kompajlerima, pa je za njih pogodno postojanje univerzalnog editora teksta (čak i ako na nekom IBM-u konstite bejzik interpretator, po izdavanju komande RUN će vam biti prijavljene sve sintaksne greške u programu što znači da se, bez vašeg znanja, vrši njegovo prevođenje). Kod kućnih računara je editoru dodata i funkcija takozvanog predprocesora (više o predprocesorima docnije), pa se on obično piše za svaki jezik posebno, pri čemu se, kod ionis ozbiljnijih računara, omogućava korisniku da bejzik program pretvori u ASCII tekst kako bi mogao da ga uklopi u neki drugi red koji priprema uz pomoć tekst procesora.

Linijski editori . . .

U doba kada su računari moćni i jeftini sa osmehom možemo da se prisetimo vremena kada su bili slabi i skupi (tako će se verovatno pisati i kroz desetak godina, pri čemu će računari našeg vremena izgledati nemoćno i skupoi). Tada je bejzik opremljen jednostavnim editorom bio prava premija: otkucate program, startujete ga i ako nađete grešku u nekoj liniji lepo je slovo po slovo prekucate, ko vam je kriv što ste pogrešili? Takav je editor (mogli bismo ga nazvati nikakvim) bio ugrađen u level I bejzik jedne od prvih mašina namenjenih širokom tržištu — računara TRS 80 model I. Kako se to široko tržište nije sastojalo od nepogrešivih, ubrzo je pripremljen level II bejzik opremljen linijskim editorom koji je, u poređenju sa nikakvim, predstavljao ogromno poboljšanje (da napravimo malu digresiju: najiskrenija tvrdnja koju je autor ovoga teksta imao prilike da pročita je sadržana u uputstvu za primenu level I bejzika. Tamo u slobodnom prevodu, piše: „zbog čega u vaš računar nije odmah ugrađen level II bejzik koji stalno pominjemo? Pa, takvo ugrađivanje svakako ne bi bio najbolji način da firma koja pravi računare zaradi pare". Ako nekada budete konstruisali računar, imajte u vidu ovu tvrdnju!)

Ima raznih linijskih editora. Najpre ćemo, potpunosti radi, pomenuti **prave linijske editore** koji su verovali ili ne, i dalje popularni kod nekih manjih velikih kompjutera kao što je PDP. Pravi linijski editor prepoznaje određeni skup naredbi koje vam omogućavaju da na ekranu dobijete određeni segment programa tj. linije sa određenim rednim brojevima — kod linijskih editora, naime, programski redovi moraju da imaju brojeve čak i kada se ne radi o bejzik programima. Te brojeve često dodeljuje sam računar po svakom startovanju editora, što znači da liniju koja je jednom imala broj 100 pri sledećoj ispravci programa može isto da ima broj 110.

Pretpostavimo da ste na ekranu dobili liniju 100 koja glasi

100 IF I>5 THEN PRINT "I>5"

Potrebno vam je, na primer, da promenite ime promenljive I u A pa ćete otkucati S(ubstitute) 100 (I/A). Linija će sada glasiti: 100 AF I>5 THEN PRINT " I '>5"

U čemu je problem? Računar je slovo I pri prvom pojavljivanju zamenio sa A čime je, umesto A>5, dobijeno AF I>5. Trebalo je, naravno, otkucati nešto poput S 100 /I/A/2 čime bismo referencirali drugo pojavljivanje slova I. Ukoliko bismo želeli da promenimo I PRINT „I>5" u PRINT „A>5" morali bismo da utvrdimo da se radi o trećem pojavljivanju slova I ili da otkucamo S 100 (I>/A). Vidimo da je rad sa pravim linijskim editorom više nego mučan: da zahteva stalnu koncentraciju korisnika i da se jednom pogrešnom komandom može osakatiti čitava linija. Jedina prednost ovakvog editora je što se komande izvršavaju tek po pritisku na RETURN, što je pogodno za komunikaciju terminala i centralnog računara.

. . . i njihove komande

Linijski editori koje poseduju TRS 80, ZX81, „spektrum", „galeksija" i neki drugi računari predstavljaju veliki korak napred. Korisnik kuca EDIT a zatim broj linije i na ekranu se pojavljuje traženi tekst. Primenom strelica ili nekih drugih specijalnih tastera kurzor se kreće po programskoj liniji što omogućava ubacivanje tekstova na željeno mesto i njegovo brisanje. Prihvatljiv editor ovoga tipa mora da ima sledeće komande:

Leva i desna strelica: pokretanje kurzora levo i desno.

BOL i EOL (Beginning Of Line&End of Line): pozicioniranje kurzora na početak i na kraj linije. Obično se za ove funkcije ne odvajaju posebni tasteri već se na početak reda pozicionira pritiskom na SHIFT i levu a na kraj pritiskom na SHIFT i desnu strelicu.

Destruktivni backspace: pritisak na njega briše slovo koje se nalazi levo od kurzora i pomera kurzor na njegovo mesto. Ostatak linije se pomera za jedno mesto levo.

Delete: briše karakter na koji je pozicioniran kurzor bez ikakvog pomeranja samoga kurzora.

Delete word: briše reč na koju je pozicioniran kurzor (od prethodnog blanka do sledećeg).

Delete end of line: briše sav tekst desno od kurzora sve do kraja linije.

Insert/overtype: tekst koji kucate može da se umeće u postojeći tekst ili može da briše tekst preko koga se kuca. Obično je editor po startovanju u insert modu ali je obezbeđena komanda koja ga prevodi u overtype.

Finish: završava editovanje linije i ubacuje rezultujući red u memoriju, prethodna programska linija sa istim brojem se uništava. Umesto specijalnog tastera za ovo se uglavnom koristi dirka RETURN (ENTER).

Quit: završava editovanje, ignorišući sve nastale promene, programska linija koja je prethodno nosila isti broj ostaje u memoriji. Obično se ovakav izlazak iz editora postiže pritiskom na BREAK (Escape).

Programiranje na sitno

Osim ovih dobri editori imaju i takozvane „undelete" komande. Kada, naime pritisnete taster Delete end of line ostatak linije ne mora da bude izgubljen nepovratno, on se jednostavno seli u neki bafer. Ako zatim primetite da ste greškom obrisali liniju pritisnućete taster Undelete eol i tekst će vam biti vraćen. Jasno je, naravno da računar ne može da pamti sva brisanja koja ste u toku jedne seanse izvršili. 'Undelete eol' je obično primenjivo samo na zadnje izvršeno brisanje.

Pomenuli smo dakle najmanje desetak specijalnih tastera koje bi trebalo obezbediti da bi jedan linijski editor mogao komforno da se koristi; gde naći mesto za njih na skučenim tastaturama kao što je „spektrumova"? Većina tastera i njihovih šifrovanih funkcija je obično zauzeta ali uvek možete odlučiti da se na primer prelazak iz insert u overtype mod vrši istovremenim pritiskom na CTRL i I. Takvo je rešenje međutim, pogodno samo za neke ređe korišćene funkcije editora: nikada ne smete sebi da dopustite da se kurzor po ekranu kreće pritiskom na SHIFT (ili CTRL) i neko slovo, ili da je funkcija Backspace ili Delete šiftovana (ako ne verujete u ovo upozorenje porazgovarajte sa nekim vlasnikom „spektruma" ili „lola 8"). Ukoliko računar koji konstruišete ima dodatnu numeričku tastaturu možete da sledite primer Digitala koji je na VAX-U ovim tasterima dodelio specijalne funkcije u toku rada ekranskog editora. Ovakvom triku, naravno, ne možete da pribegnete ako je numerička tastatura vezana paralelno standardnim dirkama sa brojevima.

Linijski editor je alatka uz čiju se pomoć može živeti sasvim lepo: ispravke programa se vrše brzo i uz minimum truda. On, međutim, ima i dve vrlo ozbiljne mane. Onemogućen je, pre svega, rad sa većim segmentima programa: ne možemo da pomerimo linije 1000—1200 tako da dođu iza linije 2300 niti da zamenimo sva pojavljivanja stringa NAME u programu sa IME. Daleko je ozbiljniji problem što se ne može editovati komandna linija. Pri radu sa kućnim računarima često smo, naime, u situaciji da izvršimo neku malu petlju punu naredbi PEEK i POKE iz komandnog moda. Pri kucanju takve naredbe koja može da bude dugačka tri ili četiri reda nije teško pogrešiti: kada nas pozdravi poruka 'Syntax error', moramo da prekucamo čitavu liniju — niko nam nije kriv što smo pogrešili. Rešavanjem ova dva problema dolazimo do ekranskog editora koma posvećujemo sledeće redove.

Ekranski editori

Razlika između dobrog linijskog i lošeg ekranskog editora je vrlo mala: otkucavši nešto poput **EDIT nnn**, dobijamo tekst linije čiji je redni broj **nnn** i taj tekst možemo da ispravljamo koristeći naredbe koje smo već upoznali. Dodatak je mogućnost da se pritiskajući gornju ili donju strelicu pomerimo na liniju koja se nalazi ispred ili iza one koju smo ispravljali i sukcesivno ponavljajući ovu operaciju pregledamo i promenimo veći deo programa. Da bi rad sa ovakvim jednostavnim ekranskim editorom bio koliko-toliko komforan, obično mu se dodaju neke naredbe kao što je pozicioniranje kurzora na početak (SHIFT i gornja strelica) i na kraj programa (SHIFT i donja strelica) te naredba koje omogućavaju brisanje čitave tekuće linije ili umetanje nove iza nje.

Ponekad je dodata i naredba SELECT koja omogućava da se određeni segment programa prenese u interni bafer odakle će biti upisan na neko drugo mesto. Tako ćemo, ako želimo da prekopiramo linije 1000—2000 iza linije 7000 najpre doći do

rednog broja 1000 onda pritisnuti dirku SELECT (npr CTRL S), a onda prelaziti kurzorom preko svih programskih linija sve do one koja nosi broj 2000 (tekst preko koga prelazimo će biti na neki način istaknut, na primer crnim slovima na beloj pozadini) Pritiskamo dirku CUT (cut-seci) i naš segment programa nestaje sa ekrana! Ne brinite, nije izgubljen pozicioniranjem iza linije 100 i pritiskom na dirku PASTE (paste-lepi) segment programa se seli na traženo mesto U SELECTovanom segmentu možemo da vršimo jednostavne pretraživanje i da zamenjujemo sva ili neka pojavljivanja jednog stringa drugim Ovakvom organizacijom nismo, na žalost, rešili problem editovanja komandne linije Pa ipak, jednostavan ekranski editor je daleko zgodniji od linijskog jer ne zahteva od korisnika koji želi da izmeni deset (obično sukcesivnih) linija programa da deset puta kuca reč EDIT praćenu rastućim nizom brojeva Činjenica da se ovaj editor primenjuje kod mnogih velikih kompjuterskih sistema i kod nekih kućnih računara nas, međutim, ne sme odvratiti od pronalaženja boljih rešenja

Karakteristike svih editora koje smo do sada upoznali je da se, za njihovo startovanje kuca reč EDIT Pre nego što otkucate ovu reč nalazite se u nekoj vrsti *nikakvog editora* koji omogućava unošenje programa i komandi, čak i ako ne biste znali da vaš računar ima neki drugi editor, mogli biste da ispravljate programe prekucavajući linije i tako se prisećate starih vremena razvoja kućnih kompjutera Vidimo da su konstruktori računara koje slede ovu koncepciju napravili dva editora od kojih je jedan vrlo jednostavan a drugi umereno složen Na užas neiskusnih korisnika može da se dogodi da ova dva editora nemaju potpuno iste komande kod galaksije taster na kome je nacrtana leva strelica u nikakvom editoru radi kao destruktivni backspace a u editoru koji se startuje sa EDIT omogućava pokretanje kurzora u okviru linije! Iako smo uz dosta muke uspeli da vam objasnimo da za svaki programski jezik treba pisati novi editor, nećemo ni pokušavati da vas ubeđujemo da ih treba pisati po dva Prosto ćemo konstatovati da se ovakvo rešenje primenjuje samo kod računara sa malim ROM-om i preći na metode realizacije ekranskih editora kod kojih je reč EDIT izgubila sva građanska prava — ovakvi su editori aktivni u toku čitavog funkcionisanja kompjutera čime je, kao što ćemo videti, najzad omogućeno editovanje komandi

Komplikovano ali ostvarljivo

„Komodor 64" je jedan od prvih širokom tržištu namenjenih računara koji je imao pravi ekranski editor Kod ovog računara, pritiskom na strelica, slobodno krećemo kurzor po ekranu i po želji na neka njegova mesta ubacujemo tekst ili brišemo postojeći Kada pritisnemo RETURN linija na koju je kurzor pozicioniran se zajedno sa svim ispravkama koje smo u nju uneli izvršava ili unosi u program baš kao da smo je u tom momentu otkucali Opisati rad „komodorovog" editora je, dakle, bilo sasvim jednostavno ali je cilj ove škole da objasnimo kako bi se sličan editor mogao napisati A to je daleko veći problem!

Pretpostavimo da naš računar ima generator karaktera tj da u tekst modu ne raspolaže nikakvom grafikom visoke rezolucije (editor može da se realizuje i bez ove pretpostavke, ali će nam ona za početak olakšati život) U startu je, dakle, u svaku ćeliju video memorije upisan broj &20 ASCII kod za blanko Kada kucamo neki tekst ili ga šaljemo na ekran primenom LIST ili PRINT naredbi,, ASCII reprezentacija tog teksta se upisuje u video memoriju, pri čemu se svaka naredba završava kombinacijom <CR><LF> (&0D &0A) Iza poslednjeg reda teksta je kurzor koji pokazuje mesto na koma će se pojaviti slovo koje korisnik otkuca, do sada je, dakle, sve sasvim jednostavno Šta se, međutim, dogoda kada korisnik pritisne neku od strelica, obično onu koja pokazuje nagore? Kurzor će se tada pomeriti za jedan red naviše i verovatno se postaviti preko nekog slova prethodne linije (čisto je tehničko pitanje kako ćete rešiti poklapanje kurzora i nekog slova ako za to postoje hardverska mogućnosti najbolje je da se kurzor ispisuje ispod slova kao zadebljana crtica koja se pali i gasi, ako hardverskih mogućnosti nema, neka se kurzor i slovo razmenjuju nekoliko puta u sekundi) Ako korisnik sada počne da kuca neki tekst, on će biti umetan na mesto kurzora a ostatak linije, zajedno sa <CR><LF> kombinacijom na njenom kraju, će polako putovati udesno Suprotno tome, ako korisnik počne da izbacuje tekst, <CR><LF> će se pokretati ulevo da bi u graničnom slučaju i ova dva karaktera bila izbrisana čime će tekuća linija biti „slepljene" sa sledećom

Pravi problem nastaje kada korisnik pritisne dirku RETURN Na računaru je tada da izvrši naredbu u okviru koje je pozicioniran kurzor ali pre toga mora da se prenese u bafer za šta je neophodno odrediti njen početak i kraj Početak nije problem treba pronaći prethodni <CR> karakter, preskočiti sledeći <LF> i tako se pozicionirati na prvo slovo naredbe (ako prethodnog <CR> karaktera nema radi se o prvoj liniji na ekranu, pa za njen početak treba usvojiti adresu prvog bajta video memorije) Kraj naredbe je daleko teže pronaći mogli bismo najpre, da u trenutku kada korisnik pritisne RETURN ubacimo u video memoriju jedan <CR> i jedan <LF> pa da te znakove smatramo krajem naredbe To, međutim, i nije tako zgodno jer od korisnika zahteva da pre pritiska na return, pozicionira kurzor na kraj naredbe koju je ispravljao, ako na to u nekom trenutku zaboravi osakatiće programski red i neće imati nikakvu indikaciju o toj svojoj grešci

Alternativa je da pritisak na RETURN ne ubaci nikakav kontrolni kod pa da računar potraži sledeći <CR> koji će biti proglašen za kraj naredbe Ovakvo se rešenje zaista i primenjuje, iako u sebi krije tragove koncepcije „dva editora za jedan jezik" kada korisnik prvi put kuca linije, pritisak na RETURN generiše <CR><LF>, kada edituje postojeći tekst, ove se kombinacije ne generišu! Kako će siroti editor znati šta da radi kada detektuje pritisak na RETURN? Uz pomoć malog trika ispitaće deo video memorije iza kurzora pa će, ako pronađe <CR>, znati gde je kraj naredbe, a ako ga ne pronađe proglasiti tekuću poziciju kurzora (ili zadnje ispisano slovo) za kraj reda i tu umetnuti <CR><LF>

Na BBC način

Ovako opisan, ekranski editor izgleda komplikovano ali ostvarljivo Ukoliko biste ga, međutim, isprogramirali i rigorozno testirali, videli biste da je podložan raznim bagovima: šta ako se korisniku prohte da pozicionira kurzor na početak ekrana na kome se već nalazi neki tekst i da onda počne da unosi nove programske redove? Posle svakoga od njih računar će pronaći po neko <CR> pa će produžiti liniju besmislicama. Čak i kada bi se tako nešto izbeglo (mogao bi se, kao što ponekad radi Hewlett Packard uvesti taster RETURN i taster EOL — oba označavaju kraj kucanja reda, pri čemu prvi uništava eventualni ostatak linije od kurzora do <CR> a drugi pokušava da pronađe bivši kraj reda; tako je čovek zadužen za rešavanje problema koji računar ne može da reši), na ekranu koji je editovan šetanjem kurzora nastaje velika konfuzija — teško je uočiti koje linije kako izgleda i slično Konstruktori ekranskog editora za „galaksiju plus" su elegantno rešili problem uvodeći tabelu dužine redova kada završimo sa editovanjem nekog reda njegova se nova dužina upisuje u tabelu i svi „repovi" nestaju sa ekrana To smanjuje nered, ali ga i dalje ne uklanja za svakog korisnika je normalno da se linija koju upravo kuca nalazi na dnu ekrana!

Konstruktori BBC računara i „electrona" su zamislili drugu vrstu ekranskog editora čije karakteristike možete da nađete u „Računarima 2", ukratko pritiskom na neku od strelica kurzor se razdvaja u dva različita kurzora od kojih jedan slobodno putuje po ekranu i pozicionira se ispod bilo kog karaktera Pritiskom na specijalni taster COPY taj karakter biva prepisan u radnu liniju a oba se kurzora pomeraju za po jedno mesto udesno Korisniku je, tako, omogućeno da sa bilo koje lokacije ekrana u radnu liniju prepisuje tekst i tako bez suvišnog kucanja povezuje nizove slova koje je sam ispisao sa nizovima slova koje je računar generisao u raznim prilikama Ovakav editor je na prvi pogled idealan ne izaziva nikakvu konfuziju na ekranu (uvek je linija koja se kuca poslednja) omogućava editovanje bilo koje od komandi i, uz poneku dodatnu opciju, olakšava rad sa blokovima programa Da stvar bude posebno lepa, ovakav editor je vrlo jednostavno realizovati, u „Računarima 4" je objavljen sličan program za „galaksiju" koji je, bez obzira na to što ga je trebalo uklapati u postojeći bejzik interpreter napisan uz utrošak svega par časova rada Obzirom da ova Škola zauzima dosta prostora u „Računarima" nismo u mogućnosti da objavimo izvorni listing ovoga programa pisan na Z80 asembleru Ukoliko vas taj listing interesuje, javite nam to pismom pa ćemo razmotriti mogućnost da vam ga pošaljemo na kućnu adresu ili da ga, ako pisama bude dovoljno, objavimo u Biblioteci programa BBC-jev editor ma koliko lepo izgledao ima i jednu veliku manu ako primetite da ste u nizu programskih redova pravili neku sistematsku grešku, moraćete da kopirate svaki od njih možda kopirate četiri reda da biste ispravili samo jedno slovo! Zato vlasnici BBC-ja često u svoje kompjutere umeću dodatne ROM-ove sa posebnim ekranskim editorima tako da za rad sa jednim jezikom imaju i po dve-tri programa za obradu teksta

Dejan Ristanović

NASTAVAK SA STRANE 11

starog dobrog Microsoftovog linijskog editora. Da dobro ste pročitali: iako je „amiga" opremljena kako mišem tako i tasterima za pokretanje kurzora, bejzik interpretator omogućava jedino ispravke programa uz primenu omražene naredbe EDIT, dok ćete, ukoliko imate običaj da kucate duge naredbe u komandnom modu, svaku grešku platiti prekucavanjem čitave linije. Uz sve ružne osobine Metacomcov bejzik je veliki potrošač memorije koji nedopustivo sporo radi: kada ga učitate, imaćete jedva 40 K (duplo manje ako se odlučite za grafiku visoke rezolucije) za bejzik programe, dok će srednje vreme izvršavanja standardnih benčmark testova 1—8 biti 5.92 sekunde (jedva brže nego na par godina starom BBC B sa osmobitnim 6502 dodatnim procesorom (9.43)! Ukoliko se na tržištu ne pojavi savremeniji bejzik (ne očekujte da će takav koštati manje od stotinak dolara), većini korisnika koji se ne usuđuju da programiraju na C-u ili asembleru će dobar deo „amiginih" fascinantnih mogućnosti ostati daleko izvan domašaja!

U cenu računara će biti uračunati i turbo paskal, logo, C kompajler i, kako to lepo zvuči, asembler i disasembler. Poseban se akcenat stavlja na C jezik na koma je uglavnom i pripreman „amigin" operativni sistem i većina raspoloživih programskih jezika. Obzirom da je tripos relativno novi operativni sistem, pre sredine iduće godine ne vredi očekivati pojavu interpretera i kompajlera drugih jezika.

Za sada je neizvesno da li će se uz „amigu" dobijati paket poslovnih programa ili će za njih biti potrebno doplata od 300 dolara. Paket bi trebalo da obuhvati solidan tekst procesor (raden, pogađate, po ugledu na MacWrite), program za crtanje (nešto kao MacPaint), program za sintezu govora i muzički procesor. Za dobar program za obradu teksta, bazu podataka i unakrsno izračunavanje (spreadsheet) treba izdvojiti otprilike dodatnih 600—1000 dolara — praktično cenu čitavog računara.

Da svedemo dakle račune: 1000 funti za računar, 180 funti za 256 kilobajta dodatne memorije, 300 funti za RGB monitor, 150 funti za još jednu disk jedinicu (računar „amigine" snage bi morao da ima bar dva flopi diska), najmanje 100 funti za dobar bejzik i još najmanje 400 funti za aplikacione poslovne programe čini oko 2100 funti za mašinu bez štampača i hard diska. Ukoliko vam ovo nije previše, „amiga" je pravi računar za vas: ni jedan od srećnika koji su imali priliku da ga testiraju nije mogao da kaže ništa osim reči pohvale, uz konstataciju da se bolje multiprogramska mašina ne može nabaviti ni za deset puta više para. Zaljubljenike u „amigu" u očajanje baca samo jedna činjenica: očekuje se da će se računar pojaviti u širokoj prodaji tek krajem ove godine, dok će prilagođenje Evropskom PAL televizijskom sistemu sačekati i prvo tromesečje sledeće. Tek ćemo tada moći da izreknemo konačnu ocenu o mogućnostima ovoga računara koji izgleda suviše dobro da bi bio — stvaran!

Dejan Ristanović

Akcije

priča o editoru

Pažljivi čitaoci su sigurno primetili u RAČUNARIMA 8 poduži spisak editorskih naredbi, na žalost bez opširnijeg komentara. Zato ćemo u ovom tekstu razmotriti, sa malo više detalja, zanimljive naredbe i mogućnosti ekranskog editora.

Da bi korisnik doveo kursor na mesto greške i izvršio ispravku, treba pre svega, dovesti „grešnu" liniju na ekran. To se može izvesti sa LIST — po želji izlistamo i nekoliko linija ispred i iza — a onda samo šetamo kursor i ispravljamo. Međutim, ako treba ispraviti samo jednu liniju, onda je EDIT naredba vrlo zgodna — jednostavno izlistava samo jednu liniju, a kursor pozicionira na njen početak. Ali EDIT naredba ima još mogućnosti: kucamo, na primer, EDIT 10,1 i na ekranu se pojavljuje deseta linija. Ako je to potrebno, vršimo neke izmene a onda pritiskamo ENTER, sada će se na ekranu pojaviti sledeće, recimo dvadesete linije programa. Ponovo po želji vršimo neku izmenu, pritiskamo ENTER i pojavljuje se trideseta linija. Tako možemo vrlo lepo da pročešljamo deo ili čak ceo program i ispravimo greške. (Neka me neko podseti na ono „spektrumovo" LIST-/BREAK/CAPS/1)

Najbolje tek dolazi

Naravno, pre nego što počne da radi, svako će poželeti da postavi boju slova i pozadine koje mu najviše odgovara, zatim da podesi brzinu pomeranja kursora, predefiniše testaturu da odgovara „spektrumu plus", odredi koliko će linija biti izlistano odjednom i možda zahteva od kompjutera da ispusti jedno „biip" posle svakog pritiska na taster. Za to služe naredbe COLOR, REP, KB-, DEFL i BEEP. (Neka me neko podseti na pokovanje sistem varijabli 23561, 23562, 23608, 23609 i slične egzibicije.) Kuriozitet je naredba FT (ili FTIME) koja određuje posle koliko vremena, ako se ne pritisne nijedan taster, kursori (glavni i kursor za kopiranje) počinju da fleširaju — kada dobijete editor, i sami ćete videti koliko je to korisno.

Stigli smo do funkcijskih tastera kojih ima tri. Zar samo tri? Da, samo tri, i to zbog organizacije testature. Pošto „spektrum" ima dva šifta — CAPS i SYMBOL — svaki taster može da ima maksimalno tri funkcije. Tako pritisak na taster C ispisuje slovo „C", CAPS/C briše ekran, a SYMBOL/C ispisuje znak pitanja. Svi tasteri zajedno sa CAPS vrše neku funkciju, a skoro svi tasteri zajedno sa SYMBOL ispisuju neki specijalan znak. Skoro svi, rekoh, jer SYMBOL/Q, SYMBOL/W i SYMBOL/E ne ispisuju ništa (<=, <> i >= se kucaju iz dva dela), pa su baš zato izabrani za funkcijske tastere. Naravno, ne bi bilo prepreka da editor ima i 150 funkcijskih tastera, ali to bi značilo povratak na onaj sistem šift prvi, šift drugi, šift treći, šift svi zajedno — nešto od čega sam hteo i uspeo da pobegnem. Uostalom, praksa je pokazala da su tri funkcijska tastera sasvim dovoljna.

Za definisanje tih funkcijskih tastera stoje dve naredbe na raspolaganju. Prva



DEFK služi za, kako bi se reklo, prosto definisanje funkcijskih tastera. Druga naredba *DEFK omogućuje da se funkcijskom tasteru dodaju i neke specijalne funkcije, kao što su pomeranje kursora, pritisak na enter i uopšte sve što se može postići pritiskom na CAPS i neki taster. Tako, ako treba da se iza desetak linije u programu dođe STOP, jednostavno se definiše jedan funkcijski taster da dovede kursor na kraj linije, otkuca STOP i na kraju pritisne ENTER.

Ako vas je sve ovo oduševilo, znajte da najbolje tek dolazi! U bilo kom trenutku pritisak na CAPS/H (help) ispisuje na ekranu koji taster čemu služi i sintaksu svih naredbi. Kada se otpusti CAPS/H, na ekran se vraća prvobitna slika — ona koju ste imali pre poziva u pomoć — pa možete mirno da nastavite svoj rad. Pretpostavimo sada da često pozivate neki potprogram

Softverski prvenci „Galaksijinih" programera dočekani su, moramo da priznamo, u izvesnoj meri uzdržar — ništa ni blizu euforiji sa računarom „galaksija" — ali ipak znatno iznad očekivanja ljudi koji su predložili i pripremili ovu akciju. Ekranski editor Vladimira Kostića, kao što se i očekivalo, pleni najviše interesovanja i predstavlja pravi hit među vlasnicima „spektruma", ali ga u stopu prati „Trodimenzionalna grafika" Jovana Skuljana. Autori svih programa intenzivno rade na ulepšavanju svojih prvenaca — Vladimir Kostić je čak, oslobođen svih autorskih obaveza prema „Računarima" samo da bi na miru ispravio poslednje bagove i napisao poslednju rutinu u svom editoru a tehničke pripreme za njihovu proizvodnju privode se kraju, pa se prvi primerci mogu očekivati sredinom decembra. Uz nekoliko novih detalja u priči o editoru, u ovom broju ponavljamo narudžbenicu i priliku da se programi nabave po povlašćenoj pretplatnoj ceni popunite dakle narudžbenicu i pažljivo očekujte poštara

Lepo je imati listing tog potprograma stalno pred očima ali nemate printer Šta sad? Izlistajte taj potprogram na ekranu i otkucajte STO (ili STORE) Dalje radite šta vam je drago a svaki pritisak na CAPS/H za trenutak prikazuje taj potprogram (Ko to još kupuje printere?) Tu su još i naredbe RCL (RECALL) i EXC (EXCHANGE) slične namene i velikih mogućnosti a ne treba zaboraviti ni FREEZE naredbu, već opisanu u prošlim „RAČUNARIMA"

Osnovni režimi

Nepravda je zaboraviti FIND naredbu Sa njom se u programu za tren oka može pronaći neka naredba ili skup znakova i po potrebi izbrisati ili zameniti nečim drugim FIND naredba kombinovana sa specijalno definisanim funkcijskim tasterima može dati izvanredne rezultate

Vlasnici mikrodrajva će sigurno blagosiljati naredbu MDF (MICRODRIVE FACILITY) neki će zavoleti JOIN ili DELETE drugima će se svideti UPPER Pošto naredbi ima zaista još mnogo bolje da pređemo na funkcije dodeljene tasterima Editor normalno radi u „overtype" modu Pritiskom na CAPS/I prelazi se u „insert" mod Umesto insert mode može se koristiti CAPS/O — jednostavno ubacuje par razmaka Za brisanja jednog znaka može da se bira između CAPS/Ø i CAPS/P, dok za brisanje od kursora pa do kraja reda služe CAPS/3 a CAPS/W briše ceo red Ubacivanje jednog praznog reda u raznim varijantama obavlja-ju CAPS/1 CAPS/2 i CAPS/O Pored standardnog pomeranja levo, desno gore, dole pritiskom na CAPS/4 moguće je dovesti kursor na početak reda CAPS/9 ga pomera na kraj reda a CAPS/V i CAPS/8 na početak odnosno kraj bejzik linije Pritisak na CAPS/Z omogućuje pomeranje alternativnog kursora za kopiranja a CAPS/X samo kopiranja Ako slučajno izgubite kursora iz vida pritisnite CAPS/SYMBOL ili SYMBOL/SPACE

Grah je na spomenuti status liniju ili recimo jedan detalj vezan za AUTO naredbu Naime u auto modu editor automatski ispisuje brojeve linije dok kucate program Ali, budite sigurni da će vas upozoriti ako neka linija već postoji — čisto da ne prekucate slučajno tu liniju i time je izbrišete iz programa

„Ponizno vas obaveštavam"

Kritičari kojima je ovaj program ne mati dele se na one koji su videli program na delu i koji su se oduševili, i na one koji ga nisu videli a apriorno kritikuju po njima enormnu dužinu pravaliki broj naredbi da možda nije dovoljno „user frendly", itd. Program jeste dugačak (u ovom trenutku 17 5K) ali to je cena za sve mogućnosti koje pruža Za korisnike ostaje sasvim prihvatljivih 23K A ruku na srce, kada ste poslednji put napisali toliki bajzik program? Što se tiče broja naredbi i funkcija, jeste da ih ima stvarno mnogo i da ih treba sve naučiti i zapamtiti, ali bolje da ih ima nego da korisnik očajava što ih nema Najzad jeste da program nije bog zna koliko „user frendly" ali namenjen je programerima a takvi kako bi to rekao Dejan Ristanović ne vole da ih mašine tretira kao budale Zar jedan editor treba da daje izveštaje poput „vrlo ponizno vas obaveštavam da je u vašem cenjenom programu došlo do jedne mizerne greščice ..."?!

Kada ću, najzad, dobiti taj editor? ! ! !" već očajavaju najzagriženiji hakeri Dok ovo pišem sve nepravilnosti u radu i meni poznati bagovi su otklonjeni Ostaje još jedino RENUMBER naredba (Kada budete čitali ovaj tekst i to će verovatno biti gotovo) Naravno, posle programskih treba rašiti sve pravnofinansijsko-štamparsko-kasetno-tehnička probleme i napisati adekvatno uputstvo za upotrebu Očekujte da do kraja godine otkucate na svojim „spektrumovima" LOAD „EDITOR" Dotle ih držite spremne i, ako još niste popunite brzo priloženu narudžbenicu

Vladimir Kostić

KAKO NABAVITI PROGRAM

U dnu ovog teksta primećujete narudžbenicu kojom možete da obezbedite svoj primerak programa ili knjiga iz naše nove Biblioteke i to u pretplati Kupovina u pretplati je kao što to obično biva povoljnija kako za vas tako i za nas vama odgovaraju niže cene a nama prilike da procenimo buduće tiraže

Da biste nabavili naslove iz naše Biblioteke jednostavno popunite narudžbenicu ili je ako ne želite da oštetite svoje „Računare" prepišite na dopisnicu i pošaljite na adresu „Galaksija" (za Biblioteku programa) Bulevar vojvode Mišića 17 Beograd Verujemo da će naručeni programi biti isporučeni do kraja godine

Obzirom da će programi 1—5 moći da se nabave isključivo posredstvom našeg časopisa svi poručioci će biti upisani u kompjutersku bazu podataka što znači da će biti lično obavešteni o unapređenjima i novim verzijama programa (dokumentacije) i da će imati priliku da te verzije dobijaju ili besplatno ili uz minimalnu uplatu

1 Ekranski editor je namenjen svima koji pišu bejzik programe. Omogućava efikasno ispravljanje softvera uz rad sa 51 slovom u redu dva kurzora funkcijske tastere prenumeraciju zamrzavanje dela ekrana različite skrolovanja i mnogo drugih stvari Ovaj 100% mašinski program je potpuno kompatibilan sa raznim proširenjima za „spektrum" Pridodata je vrlo opsežna dokumentacija Autor Vladimir Kostić

*2 Hiperbejzik je **unapređena verzija** pobednika prošlogodišnjeg „Galaksijinog" konkursa Učitavši ga bejziku vašeg „spektruma" dodajete tridesetak novih naredbi i da stvar bude posebno lepa i sami dobijate mogućnost da taj bejzik dalje proširujete bez ikakve potrebe da se bavite asemblerom Pridodata je opsežna dokumentacija rađena prema standardima našeg konkursa Autor Fischer*

3 Velika akcija je prva kompleksna igra-avantura pisana potpuno na našem jeziku Kao diverzant treba da izvedete akciju u okupiranom gradu i prebegnete na slobodnu teritoriju Program je vrlo dopadljivo realizovan tehnike kompresije ekrana su omogućile da u memoriju stane veliki broj slika u boji koje bivaju prikazane kad god dođete na neko mesto Pridodato je kratko uputstvo Autor Aca Radovanović

4 Estinglish 1 program za učenje engleskog jezika kroz igru i zabavu rađen je po principu koji su autori nazvali „All you wish" čime je program postao nešto kao korpa sa igračkama iz koje se vadi šta se želi i kada se želi Estinglish ima preko 90 slika rečnik od 360 reči kao i lekcije koje nastavnici engleskog mogu da modifikuju Pridodato je uputstvo rađeno prema standardima našeg Konkursa Autori S Milekić i D Tanaskovski

5 Trodimenzionalna grafika Jovana Skuljana — četrnaest kilobajta čistog mašinca — omogućuje grafičko predstavljanje elementarnih funkcija u tri dimenzije u aksionometriji i perspektivi Program je opremljen namenskim ekranskim editorom koji pojednostavljuje unošenje parametara i da bi se dobilo na brzini prikaza sopstvenim kalkulatorom

NARUDŽBENICA

Ovim neopozivo naručujem pouzećem sledeće programe iz „Galaksijine" biblioteke

1 Ekranski editor	(spektrum)	800 din
2. Hiper bejzik	(spektrum)	700 din
3. Velika akcija	(spektrum)	400 din
4. Estinglish	(spektrum)	800 din
5. Trodimenzionalna graf	(spektrum)	600 din

Takođe naručujem sledeće knjige

5. Sve „spektrumove" rutine	380 din
6. Sve „komodorove" rutine	380 din

Ime i prezime ____________________

Adresa ____________________

Mesto ____________________

Potpis ____________________

Odgovarajući iznos ću uplatiti poštaru prilikom preuzimanja programa

Makazama po ROM-u

gens iz roma

ZX Spectrum

U „Računarima 8" („ROM od sedam milja") bio je objavljen projekat za proširenje „spektrumovog" ROM-a. Svaki komercijalni program mogao se ubaciti u jedan EPROM i tako čuvati u računaru dok ga ne pozovemo. U trenutku poziva, program se preslikava na svoje uobičajeno mesto u RAM-u, gde započinje njegovo izvršavanje, baš kao da je učitan sa kasete. Jednostavno rečeno, ROM od „sedam milja" je jedan super brzi kasetofon — brži čak i od famoznog hard diska.

Tada smo još mislili da dalje nema smisla ići. Prepravljati programe tako da rade neposredno iz ROM-a, zvučalo je ako ne nemoguće ono bar neracionalno u doslovnom smislu te reči. Međutim, danas o tome imamo sasvim drugo mišljenje. Programi koji se ne oslanjaju previše na osnovni ROM mogu se preraditi tako da rade umesto njega. Pred nama je, evo, DEVPAC GENS asembler prilagođen radu iz ROM-a, a da čitav posao oko toga nije predstavljao baš nikakav istinski problem (izuzev ako u problem ne uračunamo veliki broj utrošenih časova rada). I pošto, kao i obično, svoj uspeh želimo da podelimo sa čitaocima „Računara", odlučili smo da opišemo ukratko ideju za prepravku, kako bi svi oni koje to interesuje mogli i sami da prilagode svoj asembler.

Ukoliko, naravno, iz bilo kojih razloga, nemate nameru da se upuštate u ovu avanturu, a ipak biste želeli asembler u ROM-u, verujemo da vam neće predstavljati veliki problem da odvojite malo novca i pošaljete svoj EPROM ili kasetu na adresu naše redakcije, gde ćemo vam obaviti programiranje.

Podrazumeva se, svakako, da je vaš „spektrum" već podvrgnut operaciji proširenja ROM-a, jer bez te hardverske izmene sve što dalje budemo govorili nema nikakvog smisla.



Šematski prikaz programa GENS3M

BLOK 1	&7530	Program za relokaciju (29)
BLOK 2	&754D	Glavni program GENS3M (6370)
BLOK 3	&8E2F	Tabele koju koristi asembler pri prevođenju (620)
BLOK 4	&909B	Adrese za ulaz u prethodnu tabelu (18)
BLOK 5	&90AD	Adrese izvršnih rutina naredbi editora (57)
BLOK 6	&90E5	Adrese izvršnih rutina naredbi u E modu (42)
BLOK 7	&9110	Sistemske promenljive (132)
BLOK 8	&9194	Tekstovi poruka (362)
BLOK 8	&92FE	Relativni položaji adrese za relokaciju (1418)

Kako radi GENS?

Program GENS3M, sa kojim smo se mi odlučili, sadrži ukupno 9046 bajtova. Koncipiran je tako da se može startovati sa bilo koje adrese, ali isključivo u RAM-u jer neki njegovi delovi u toku rada trpe izmene: procesor u njih nešto upisuje.

Da bismo konkretizovali stvar, uzećemo da je GENS unet na adresu 30000 (&7530). Startovanje se u tom slučaju vrši sa **RANDOMIZE USR 30000.**

Prva stvar koju GENS neizostavno mora obaviti jeste *relokacija* svih adresa, odnosno njihovo prilagođavanje početnoj adresi &7530. Naime, u početnom stanju, svuda u programu gde treba da stoji apsolutna adresa (recimo iza **JP** i **CALL** instrukcija), stoji ustvari *relativna* adresa, merena u odnosu na prvi bajt programa. Tek kada se program unese u memoriju računara, i kada postane poznata početna adresa (&7530), moguće je zameniti relativne adrese apsolutnim.

Da bi znao gde se sve u okviru njega nalaze adrese koje treba relocirati, GENS koristi jednu tabelu sa samog kraja programa (poslednjih 1418 bajtova između adresa &92FE &9885). Tu se nalaze položaji (opet

Ako pišete programe na mašinskom jeziku, onda svakako znate da se taj posao ne može praktično ni zamisliti bez asemblera — posebnog sistemskog programa koji služi za unošenje i prevođenje mašinskih naredbi. A ako imate asembler, onda znate i to da svako njegovo učitavanje sa kasete ne traje manje od jednog minuta, što može postati veoma neprijatno kada se mora ponoviti više puta u toku istog dana. Čak i da nam to ne smeta, ostaje još uvek prisutan veliki nedostatak ovog, i njemu sličnih programa bar 8K memorije troši se na njegovo smeštanje, dok pri tome ROM ostaje uglavnom neiskorišćen (asembler ne koristi bejzik interpreter). Kada bi se asembler nalazio u ROM-u, ni jedan od ovih problema ne bi postojao Poziv programa bi se obavljao trenutno, jednom ukucanom naredbom, a čitav raspoloživi RAM bio bi otvoren za naše potrebe

relativni) za 707 adresa koje se moraju relocirati Poslednji podatak u tabeli je &0000 i prosto označava kraj

Program za relokaciju počinje na &7530 i zauzima 28 bajtova (zaključno sa adresom &754C) Kad se posao jednom obavi skok se vrši na adresu &7572, gde u suštini počinje sam GENS (inicijalizacija) Program i tabele za relokaciju više se nikad neće koristiti Uostalom, tabele će ubrzo biti prebrisane mašinskim stekom, baferom i, naravno, samim tekstom koji budemo ukucali

Jasno je sada zašto se GENS ne sme ponovo startovati od adrese &7530 Na raspolaganju nam stoje dve druge adrese &7568 za hladan i &756D za vruć start, ali je sasvim izvodljivo pozvati i &7572, što će dovesti do pravog ponovnog startovanja, sa porukom „Buffer size?"

GENS3M dakle efektivno zauzima adrese između &754D i &92FD Pri tome počev od &&E2F, prostiru se same tabele, tako da se čist program završava na &8E2E U šemi koju dajemo prikazana je struktura čitavog programa Blokovi su označeni rednim brojem i početnom adresom, a u zagradi je data dužina bloka u bajtovima

Kako disasemblirati GENS?

Prvo što se mora uraditi sa programom koji želimo da prepravljamo jeste njegovo disasembliranje Treba sesti za računar i učitati GENS3M počev od adrese 30000, a zatim i MONS3M počev od 55000

Zatim je neophodno obaviti relokaciju GENS-a. Međutim, da bismo izbegli pri tome i njegovu inicijalizaciju, zamenićemo naredbu **JR Z,&7572** na adresi &753A sa **RET Z** i **NOP.** To sve, naravno, radimo pomoću MONS-a Tek nakon toga izvršimo iz bejzika **RANDOMIZE USR 30000,** čime je relokacija obavljena

Disasembliranje počinjemo od adrese &754D, koristeći opciju „$", i pri tome *prepisujemo na papir naredbu po naredbu* Biće nam potrebna sveska A4 formata od stotinak listova, i nekoliko dana ne previše napornog rada (po par sati dnevno) Bilo bi, svakako, jednostavnije koristiti naredbu „T", uz mogućnost upotrebe štampača, ali nema nikakvog smisla prevoditi velike blokove programa čiju strukturu zapravo i ne poznajemo Disasembler, sam po sebi, ne zna da li prevodi naredbe ili obične podatke iz neke tabele

GENS3M sadrži u sebi nekoliko tabela, na koje ćemo u procesu disasembliranja naići To će se desiti na adresama &7562—&7566, &7611, &7698 — &76AD — &76E1 i &7730 — &7739. Osim toga, nekoliko puta će se pojaviti naredba **RST &08, iza koje obavezno mora doći jedna DEFB.**

Kada disasembliramo nepoznati program najbolje je pratiti samo njegov tok. Time izbegavamo mogućnost da upadnemo u neku tabelu pa da nju počnemo da prevodimo A ako se to ipak i desi, sigurno ćemo primetiti besmislen niz naredbi, kao signal da smo zalutali

---

Servis za eprome

Objavljivanje kompletne dokumentacije za prepravku GENS-a, kao i za prepravku osnovnog ROM-a, stavilo bi na veliku probu strpljenje onih čitalaca koje ove (uzbudljive) teme možda nimalo ne interesuje Stoga smo odlučili da, po simboličnim cenama, organizujemo servis za programiranje eproma Nove verzije osnovnog ROM-a, sa ispravljenim bagovima u operativnom sistemu i operativnim sistemom za rad sa alternativnim ROM-ovima, kompatibilna je sa svim programima koji ne koriste interfejs 1. Uz programirane ROM-ove biće priložena i sva dokumentacija koja je neophodna za nesmetano korišćenje i programiranje nove verzije računara

Vlasnici „spektruma" mogu da se odluče za jednu od sledećih varijanti:

- uparena verzija osnovnog ROM-a i GENS-a (dva eproma 27126) 600 din
- programiranje eproma sa GENS-om 400 din
- programiranje eproma sa osnovnim ROM-om 400 din
- brisanje eproma 200 din
- verzija na kaseti (sa kasetom)

Kod varijanta sa programiranjem potrebno je na adresu redakcije — „Galaksije" — BIGZ, 11000 Beograd, Bulevar vojvode Mišića 17 — poslati jedan ili dva eproma tipa 27128 i potvrdu o uplati odgovarajuće sume, a oni koji žele sami da programiraju svoje eprome treba da dostave redakciji samo uplatnicu za verziju na kaseti Programiranje eproma ne može trajati duže od deset dana Svaki eprom koji se — krivicom redakcije — zadrži u „Galaksiji" duže od ovog roka biće programiran potpuno besplatno

---

Posle manje ili više teškoća, disasemblíraćemo tako čitav kôd između &754D i &8E2E Međutim, ako nismo obraćali pažnju na smisao naredbi, već samo na njihovo prepisivanje, u programu će nedostajati veliki broj labela na koje se obavljaju skokovi ili na kojima počinju razni potprogrami Moramo, zato, sada pročitati ceo program ponovo i potražiti naredbe za skok, dopisujući labele tamo gde nedostaju Takođe se moraju dopisati i labele utabličene u blokovima 4, 5 i 6 I naravno, dok čitamo program, moramo obratiti pažnju na sve one blokove u koje se u toku rada programa nešto upisuje Takvi blokovi se ne mogu naći u ROM-u, pa ih moramo odvojiti od ostalog programa Ukoliko nismo sigurni da smo otkrili sve „promenljive" bajtove, možemo jednostavno startovati GENS, nešto s njim raditi a onda se vratiti u bajzik i obaviti poređenje korišćenog i nekorišćenog (ali relociranog) programa Najjednostavnije je, pri tome, koristiti naredbu **VERIFY,** pri čemu prethodno treba snimiti na kasetu čist relociran GENS3M

Tek sada možemo upotrebiti opciju „T" iz MONS-a Disasembliranje ćemo vršiti u blokovima od oko 1K, što će dati tekst dužine 8K približno Blokove snimamo na kasetu a kasnije ih povezujemo u dve veća (od po 24K) Razume se, povezivanje vršimo pomoću GENS-a, a takođe dopisujemo i sve nedostajuće labele poredeći tekst sa našom sveskom Konačno sve apsolutne adrese moramo zameniti simboličkim, najjednostavnije pretvaranjem znaka „ " u „L" ispred adrese Tu će nam dobro poslužiti naredba „F" u okviru GENS-a ali moramo paziti da ne pređu baš svi „ " u „L", što znači da treba opet sve pratiti po onoma što smo zapisali u svesci

Tu je, otprilike, kraj svih problema Dodaćemo tekstu jednu ORG naredbu i prosto asemblirati program tako da radi sa izabrane adrese Povezaćemo to sa neophodnim tabelama i ceo kôd ubaciti u EPROM odakle će se izvršavati

Šta mora ostati u RAM-u?

GENS3M je dosta dobro organizovan, jer su bezmalo sve sistemske promenljive smeštene zajedno u jedan blok (7 na našoj šemi). Međutim, postoji nekoliko promenljivih u bloku 2 &7562—&7566, &7611 i &76DE—&76E1 Svi ti bajtovi, razume se, moraju ići u RAM, a to se odnosi i na sve adrese posle &92FE (blok 9) Ostali delovi programa su nepromenljivi i mogu se ubaciti u ROM, a ukupna dužina im iznosi 7516 bajtova.

U istom EPROM-u moraju se naći i svi potprogrami koje GENS normalno poziva iz „spektrumovog" ROM-a. To su rutine KEYBOARD (&02BF), BEEPER (&03B5), SA BYTES (&04C2), LD BYTES (&0556), PRINT A (&0010) i CHAN OPEN (&1601) Ove rutine se pripremaju na isti način kao i GENS Uz malo preuređenja, biće se sve dovoljno mesta

Da bi startovao GENS, treba samo preneti kontrolu na odgovarajući ROM i skočiti na potrebnu adresu Pri povratku u bejzik mora se, naravno, uključiti osnovni ROM Sve te poslove može da obavlja jedan mali mašinski program smešten negde u RAM-u Detalje oko praktične realizacije prepuštamo vama.

Čitaoci koji budu poručili program od nas dobiće kompletno uputstvo, sa mapom memorije i organizacijom novog ROM-a. A ako radite sami, pa naiđete na problem, tu smo da pomognemo

Jovan Skuljan

kvadratni koren (i)

To može i bolje (I)

Osmobitni računeri obično vrednost kvedratnog korena Y izračunavaju prema formuli

$$Y_e = EXP(0.5*LOG(X))$$

Funkcija LOG (prirodni logaritam) I EXP svejedno postoje u računaru, pa ovaj postupak zahteva malo memorijskog prostora. Da li je svođenjem na stepenovanja problem rešen? To samo tako izgleda. Ovako rešenja se ne može prihvatiti već zbog toga što je kvadratni koren daleko najčešća funkcija, pa njeno izračunavanje mora biti tačno i brzo. Navedenom formulom za male pozitivne i za velike vrednosti argumenta ne dobije se tačne vrednost: čak sedem bitova mantisa mogu biti netačni (i onda kada su funkcija LOG I EXP besprekorno urađene). Osim toga, izvršavanje programa LOG I EXP su dugotrajni, pa se nedovoljno tačan rezultat Y_e dobije za nedopustivo dugo vreme.



Predstavljanje brojeva

Brojevi se u osmobitnom računaru obično predstavljaju se

$$X=0 \text{ ili } X=\pm 2^k m,$$

gde je K ceo broj (karakteristike) i m — normalizovane mentise $0.5<=m<1$. Kod osmobitnih računera karakteristika uvećane za 128 obično se smešta u prvi bajt broja. Ostali bajtovi služe za smeštaj J bitova mantise, s tim što se umesto prvog bita (koji je uvek 1) beleži znak broja (0 za x>0, ili 1 za x<0). Broj bitova rezervisanih za smeštaj karakteristika je odgovoran za opseg brojeva $-127<=K<=127$, pa je $X=0$, ili $2^{-128}<=X<2^{127}$. Broj bitova rezervisan za smeštaj mantise J određuje tačnost predstavljanje promenljive 2 J. Vrednost funkcija treba da bude isto toliko tačne. Na sl. 1 prikazana je relativna greška vrednosti kvadratnog korena kada se računa onako kako to čine mnogi osmobitni računari.

Ovakav način beleženja brojeva imaju, na primer, komodor 64, spektrum i šarp (u HuBASICu). Računar „galeksija", međutim, beleži brojeve nešto drukčije.

Korekcija firminih programa

Poboljšanje tačnosti firminih programa može se postići formulom koju je zneo Heron još pre dva hiljade godine

$$Y_n=(X/Y_{n-1}+Y_{n-1})/2$$

Dovoljna je samo jedna iteracija da se postigne tačnost računara. Razume se da ta jedna iteracija produžava izračunavanje već ionako sporog algoritma, ali tek se njom u promenljivoj Y imamo očekivanu vrednost. Ovim je, za one koji štede na memorijskom prostoru i ne štede na vremenu izračunavanja programa, problem rešen

```
                    ZI SPECTRUM u/c

RST               Rutina je ralunatisina  Kursor: I P  Calculator
                  i njegova memorija M0  Na ulazu je potrebno da
                  se na vrhu računskog steka nalazi broj, a pri
                  izlazu na vrhu biće njegov kvadratni koren
RST 028           Ulazak u kalkulator  Na njegovom steku je x
DEFB 031          Kopiranje broja sa vrha steka  Na steku: x,x
DEFB 031          Na steku: x,x,x
DEFB 037          Da li je X>0 ? Na steku: x,x,1/0
DEFB 0,7          Ako je 1 skok za 7 bajtova  Na steku: x,x
DEFB 038          Negacija. Ako je x=0, negacijom postaje x=1
DEFB 0,26         Ako je na vrhu steka 1, skok za 26 bajtova (na
                  poslednji DEFB 038)
DEFB 038          Izlazak iz kalkulatora, samo ako je na ulazu
                  bilo x<0  Argument je pogrešan
RST 8             Skok na ERROR rutinu (iz ROMa)
DEFB 9            Kod greške "A Invalid argument"  Program se
                  prekida i vraća u bejzik editor
DEFB 006          Pravođenje u realan broj  Početak računanja
                  kvadratnog korena za x>0  Na steku: x,x
DEFB 031          Na steku: x,x,x
DEFB 031          Na steku: x,x,x,x
DEFB 001          Na steku: x,x,x,x,x
DEFB 038          Izlazak iz kalkulatora
LD A,(HL)         U akumulator se smešta vodeći bajt broja
                  (karakteristika uvećana za 128)
SRL A             Deljenje vrednosti u akumulatoru na dva
                  Delljenjem, uz postavljanje Carry flag u na 1
                  ako je karakteristika bila neparna
ADC 040           Na vrednost u akumulatoru dodaje se 128 i
                  vrednost Carry flag-a.
LD (HL),A         Broj iz akumulatora upisuje se kao vodeći bajt
                  broja na vrhu računskog steka  Sada je na vrhu
                  steka približna vrednost kvadratnog korena
LD B,4            U registar B se unosi broj iteracija
RST 028           Ponovni ulazak u kalkulator  Na steku: x,x,x,x,y
DEFB 0C0          y je zapamćeno i u memoriji M0  Na steku u prvom
                  prolazu: x,x,x,x,y, a u poslednjem: x,y
DEFB 005          Na steku: [illegible]
DEFB 0E0          Na steku: [illegible]
DEFB 00F          Na steku: [illegible]
DEFB 0A2          Na steku: [illegible]
DEFB 004          Na steku: [illegible]
DEFB 035,0F5      Vrednost u registru B se smanjuje za 1 i ako je
                  rezultat 0 iteracije su završene, a ako nije
                  skok na DEFB 0C0.
DEFB 038          Izlazak iz kalkulatora
RET               Povratak iz potprograma  Na vrhu steka se nalazi
                  vrednost kvadratnog korena
```

za svako X>0. Evo kako bi izgledao kompleten bejzik program

```
10 INPUT X: LET Y=SQR(X)
20 IF X=0 THEN GO TO 40
30 LET Y=(X/Y+Y)*0.5
40 PRINT Y
```

Na primer, na spektrumu se program SQR računa za 116 ms (milisekundi), a jedna iteracija oko 15 ms, što ukupno iznosi oko 133 ms!

Samo Heronov obrazac

Ako imamo dovoljno memorijskog prostora i važna nam je i brzina izračunavanja programa, sigurno ćemo se opredeliti za algoritam i program koji ne poziva funkcije LOG i EXP.

Heronov obrazac konvergira za svaku pozitivnu vrednost početne aproksimacije Y_e. To ne znači da treba vrednost početne aproksimacije Y_e prepustiti slučaju — naravno da ne želimo stotina iteracija.

Za početnu aproksimaciju Y_e u literaturi se sreću, između ostalog, i ove vrednosti

$$Y_e=1 \text{ ili } Y_e=X \text{ ili } Y_e=X+1$$

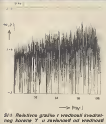
koja dovode do rezultata ali uz veliki broj iteracija. Broj potrebnih iteracija (n) može da se izračuna iz formule

$$n=INT(ABS(K)/2)+5$$

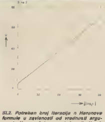
Sa INT(z) označena je funkcija najveće celo koja ne premaša z. Kako je $-127<=k<=127$, potrebno je do 69 iteracija. To je nedopustivo veliki broj iteracija. Grafička interpretacija poslednje formule data je na sl. 2.

Način zaustavljanja pri iterativnim računima je veoma bitan. Ma koji broj da se usvoji za grešku ili moduo greške koja se toleriše, postoji problem: za neke argumente bio bi to strog uslov (koji se ne može ispuniti), za druge argumente uslov bi bio suviše blag (koji je ispunjen mnogo pre nego što se dobije prva tačna cifra rezultata). Jedino ispravno stanovište je da se zahteva tačnost do poslednjeg bita mantise. To je mogućno postići i to treba postići

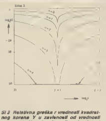
Izračunavanje vrednosti kvadratnog korena je problem na kome se može veoma lako uočiti razlika između korisnika računara i programera. Kaorisnik računara zna da je računar snabdeven programom SQR, koji za zadati argument X daje vrednost kvadratnog korena SQR(X). Programer zna više od korisnika računara — zna koje su slabosti firminog algoritma, zbog čega u nekim intervalima argumenta program veoma dugo radi, ili daje nedovoljno tačne rezultate. Programar zna i bolje algoritme koji obezbeđuju tačnost izračunavanja vrednosti kvadratnog korena do poslednjeg bita, pri čemu se rezultat dobija veoma brzo, sa svega nekoliko iteracija. U seriji napisa „To može i bolje" objavićemo algoritme za izračunavanje vrednosti osnovnih funkcija. Ovi algoritmi predstavljaju originalne rezultate gotovo dvadesetogodišnjeg istraživačkog rada prof. dr Dušana Slavića u oblasti numeričke analize.



Sl 1. Relativne greške r vrednosti kvadratnog korena Y u zavisnosti od vrednosti argumenta X ako se koristi samo formula Y_e=EXP(0.5*LOG(X))



Sl.2. Potreban broj iteracija n Heronove formule u zavisnosti od vrednosti argumenta X za početne aproksimacije Y_o=1, Y_o=X i Y_o=X+1



Sl 3. Relativna greška r vrednosti kvadratnog korena Y u zavisnosti od vrednosti argumenta X posle n iteracija, pri početnoj aproksimaciji $Y_o = 2^{INT((-K+1)/2)} m$, I je ceo broj.

Početna aproksimacija

Broj iteracija se radikalno smanjuje ako se vodi računa o načinu predstavljanja brojeva u računaru

Testiranje novog kvadratnog korena se može obaviti sledećim programom:

```
10 LET I=0: LET Z=0: LET Y=0
20 INPUT Z
30 RANDOMIZE USR 56000
40 PRINT Y,(SQR(Z) Y)/(Y+1E-38)
```

Njime se, za zadatu vrednost Z, dobijaju tačne vrednosti kvadratnog korena, kao i relativna greška Spektrumovog SQR(a). Mašinski program na adresi 56000 je sledeći:

```
ORG 56000   Startna adresa je proizvoljna
CALL PUG    Postavljanje argumenta (prve promenljive u
            bejziku programu) na vrh računskog stala
CALL SQRT   Rutina novog programa za računanje kvadratnog
            korena.
CALL SGT    Premeštanje rezultata sa vrha računskog stala u
            bejzik (na mesto treće promenljive u programu)
RET         Povratak u bejzik
```

Potprogramm PUG i SGT su dati u članku "Mašinska veza" (RAČUNARI br 8). U tom članku je opisan i način njihovog korišćenja.

U literaturi se navodi početne aproksimacije

$$Y_o = 2^{INT(K/2)}$$

Programski je jednostavnija aproksimacija

$$Y_o = 2^{INT((K+1)/2)} m$$

Ta formula je korišćena u programima pisanim za „komodor C-64" ZX „spektrum" i „šarp". Sva tri programa sadrže test da li je argument negativan — tada treba javiti grešku. Za X=0 rezultat je Y=0. U svim ostalim slučajevima dovoljne su četiri iteracije za standardnu tačnost od 32 bita mantise (skoro 10 značajnih cifara)

```
1 REM     COMMODORE 64
10 X=0 : Y=0
20 INPUT X
30 Y=X : IF Y>0 GOTO 60
40 IF Y=0 GOTO 90
50 PRINT"Illegal argument":STOP
60 I=256*PEEK(46)+PEEK(45)+9
70 POKE I,INT((PEEK(I)+129)*.5)
80 FOR I=1 TO 4:Y=(X/Y+Y)*.5:NEXT
90 PRINT Y
```

```
1 REM     ZX SPECTRUM
10 LET X=0: LET Y=0: LET P=1/2
20 INPUT X
30 LET Y=X/1:IF Y>0 THEN GO TO 60
40 IF Y=0 THEN GO TO 90
50 PRINT"Illegal argument":STOP
60 LET I=256*PEEK(23628)+PEEK(23627)+7
70 POKE I,INT((PEEK(I)+129)*P)
80 FOR I=1 TO 4: LET Y=(X/Y+Y)*P:NEXT I
90 PRINT Y
```

```
1 REM     SHARP MZ-700 (HuBasic)
10 DEFDBL A-H,O-Z
20 INPUT X
30 Y=X : IF Y>0 GOTO 60
40 IF Y=0 GOTO 90
50 PRINT"Illegal argument":STOP
60 I=VARPTR(Y)
70 POKE I,INT((PEEK(I)+129)*.5)
80 FOR I=1 TO 5: Y=(X/Y+Y)*.5: NEXT
90 PRINT Y
```

Kod računara „spektrum" postoji greške pri zadavanju broja 0.5. Binarni zapis tog broja u memoriji računara je pogrešan jer algoritam za deobu nije besprekoran, vrednost količnika 1/2 je tačna, pa je to i iskorišćeno.

Računar Šarp sa HuBASICom ima mogućnost i dvostruke tačnosti od 56 bitova (oko 17 značajnih decimalnih cifara). Za dvostruku tačnost potrebno je pet iteracija samo jedna više nego za standardnu tačnost. U tabeli 1 je prikazan broj tačnih bitova za dati broj iteracija i navedenu početnu aproksimaciju

TABELA 1

Broj iteracija	1	2	3	4	5
Broj tačnih bitova	4	9	19	39	80

Relativna greške vrednosti kvadratnog korena u zavisnosti od vrednosti argumenta, pri raznom broju iteracija, data je i na sl 3.

Da bi se dobila mogućnost za procenu brzine ovog algoritma urađen je mašinski program za računar „spektrum". On se sastoji od 36 bajtova i oni koji žele da menjaju „spektrumov" ROM mogu ga uneti u prazne memorijske lokacije. Sve što je potrebno dodati je izmena u tabeli adresa u kalkulatoru, koja bi pokazivala na adresu nove rutine. Za korišćenje rutine SQRT iz bejzika potreban je još jedan mašinski potprogram koji omogućuje komunikaciju mašinske sekvence sa bejzik programom.

Kada je računanje kvadratnog korena u pitanju analiza pokazuje da je „spektrum" znatno brži sa novom rutinom. „Fabrički" SQR se računa za 118 ms, a novi SQRT za 24.6 ms.

Možda se nekom korisniku računara (koji neosnovano veruje da je programer) ovaj tekst učinio previše stručnim. Tu nema pomoći! Ima još mnogo znanja i umeća kojima korisnik računara treba da ovlada ako želi da postane programer. Razumeti probleme i kako se rešavaju pri izradi algoritma i programa za računanje kvadratnog korena predstavlja tek prvi korak.

Dušan Slavić, Ninoslav Čabrić

mašinska veza II

Pozivanje mašinskih programa iz bejzika

U „Računarima 3" bilo je dosta reči o tome kako preneti parametre iz bejzika u mašinski program. Rešenje je bilo u definisanju jedne **DEF FN** funkcije sa onoliko argumenata koliko ima i ulaznih parametara u mašinsku rutinu. Ako, na primer, program na adresi 50000 zahteva na ulazu dva broja i jedan znakovni niz, uredićemo

```
10 DEF FN p (x,y, a$) = USR 50000
```

Mašinski program će dalje lako prikupiti parametre, koristeći sistemsku varijablu DEFADD. Recimo, za ulazne podatke 2,5 i „Beograd" dovoljno je u bejziku izvršiti **RANDOMIZE FN p (2,5, „Beograd").**

Prikupljanje rezultata

Rešenje je jednostavno, pregledno i idealno za sve mašinske programe koji ne vraćaju nikakve izlazne parametre.

Problemi nastaju kada mašinski program ima da vrati bejziku neke izlazne veličine. To mogu biti rezultati izračunavanja ili bilo kakvih operacija nad brojevima i nizovima.

U „Računarima 8" data je ideja koja je primenljiva na sve mašinske programe, sa proizvoljnim brojem ulaznih i izlaznih parametara, mada je pažnja posvećena samo numeričkim parametrima. Jedino što u svemu tome možda može da zasmeta jeste činjenica da svi parametri zahtevaju fiksne bejzik promenljive preko kojih se obavlja prenos. Naravno, vlasnici „spektruma" su odavno navikli da podnose mane svog računara pa im ni ovo ograničenje ne pada previše teško.

Ipak, potražićemo i neka druga rešenja za prikupljanje izlaznih parametara, a ograničićemo se na tri najjednostavnija slučaja:

- rezultat je ceo broj između 0 i 65535,
- rezultat je realan broj u obliku pokretnog zareza, i
- rezultat je alfanumerički niz.

Slučaj celog broja već je bio rešen. Pošto funkcija **USR**, pa prema tome i odgovarajuća **DEF FN** funkcija, po prirodi stvari ima 16-bitni rezultat, dovoljno je, pre povratka u bejzik, ubaciti rezultat u registar BC. Primer je bila funkcija za sabiranje

```
10 DEF FN f (x,y)= USR 23296
```

Zbir dva cela broja, recimo 7 i 9, neposredno se dobija sa **PRINT FN f (7,9).**

Realan rezultat

Tehnika sa registrom BC, zahvaljujući upravo ograničenjima funkcije USR, ne pomaže ako mašinski program treba da nam vrati neki broj u pokretnom zarezu. Da bismo, međutim, razumeli o čemu se radi, moramo dobro poznavati „spektrumov" ROM i to naročito deo za vrednovanje aritmetičkih izraza.

Prvo pitanje bilo bi: gde se to obavlja povratak iz našeg mašinskog programa, tj. kuda odlazi računar kada naiđe na našu **RET** naredbu? Svakako, povratak se vrši u bejzik, ali to ništa konkretno ne znači. Na koju adresu u ROM-u se prenosi dalje izvršenje programa? Nakon našeg **RET**, „spektrum" se vraća uvek na adresu &2D2B, a tamo počinje zapravo rutina STACK_BC, koja prosto uzima sadržaj registra BC i smešta ga na računski stek, formirajući tako rezultat funkcije **USR**. To smo skoro mogli i očekivati jer „spektrum" nalazi rezultate svih funkcija i operacija upravo na računskom steku.

Ideja bi bila sada sledeća: sami ćemo ubaciti na računski stek neki svoj broj, a ubedićemo računar da ne izvrši STACK_BC posle našeg **RET**. Na taj način bi funkcija **USR** bila „izračunata" sa uspehom kao i obično, ali bi njen rezultat, u stvari, bio broj koji smo mi pominjali.

Ubeđivanje računara je tu možda i najlakši deo posla. Prosto treba pogledati mašinski stek jer se tamo inače nalaze sve adrese za povratak iz potprograma. Sama adresa &2D2B biće u vrhu steka čim počne da se izvršava naš mašinski program. To je jasno jer računar će, kad bude naišao na **RET**, skinuti upravo poslednju adresu sa steka i obaviti povratak na nju. Mi, dakle, samo treba da uklonimo tu adresu, recimo jednim **POP HL**, najbolje na početku programa da posle o tome ne mislimo.

Sigurno se pitate šta će se sada dogoditi? Na mašinskom steku više nema adrese &2D2B. Da li će se povratak u bejzik uopšte uspešno obaviti?

Odgovor je: da i oko toga nimalo ne treba brinuti. Povratak će se, umesto na &2D2B, obaviti na &3365 (RE_ENTRY u okviru CALCULATE). Najjednostavnije rečeno, čim izračuna funkciju **USR** (sa podmetnutim rezultatom) „spektrum" nastavlja sa daljim računanjem izraza u okviru koga se **USR** i našlo. Jedino se mora zapamtiti da osim uobičajenog stanja registra HL, koji na povratku mora sadržati &2758, sada i IY, ukoliko smo ga dirali, moramo vratiti na &5C3A sami, jer je o tome inače brinula rutina STACK_BC koju smo mi sada zaobišli. I konačno, sve ovo neće imati nikakvog smisla ukoliko zaboravimo da postavimo na računski stek svoj rezultat za **USR**. Kada uzima neki broj sa računskog steka, „spektrum" uopšte ne proverava da li tamo nečeg ima ili ne. A skidanje sa praznog steka dovešće do velikih problema.

U primeru 1 dajemo kako bi izgledao program za sabiranje realnih brojeva. Mi smo izabrali adresu 50000, ali to se, jednostavno, može izmeniti korigovanjem **ORG** naredbe. U bejziku, naravno, treba ukucati liniju

```
10 DEF FN r (x,y)=USR 50000
```

Tako, na primer, da bi se dobio zbir brojeva 3.27 i 8.549 treba izvršiti **PRINT FN r (3.27,8.549).** Jasno, sabiranje brojeva u bejziku može se ostvariti na daleko jednostavniji način, ali nama je jedino bio cilj da objasnimo jedan metod čija je primena, razume se, van granica osnovnih aritmetičkih operacija.

Alfanumerički rezultat

Preostaje još slučaj kada želimo da mašinski program vrati bejziku rezultat tipa znakovnog niza. Uzećemo za primer funkciju koja prihvata proizvoljan alfanumerički niz, a vraća taj niz poređan u opadajućem poretku, pri čemu se porede ASCII kodovi znakova.

Na izgled, ništa posebno tu nema da se kaže. Treba napraviti program za uređenje niza i funkciji **USR** podmetnuti odgovarajući rezultat. U dodatku „Sve spektrumove rutine" objasnili smo da na računskom steku i nizovi zauzimaju pet bajtova, mada se tu ne čuva sam niz već njegove početne

Primer 1

```
        ORG     50000
SUMA    POP     HL             Uklanja se sa steka adresa STACK_BC
        LD      HL, (&5C0B)    Uzima se iz DEFADD adresa parametara
        INC     HL             i vrši se pomeranje
        INC     HL             na početak prvog parametra (x)
        CALL    &33B4          Pozivom STACK_NUM parametar se
                               šalje na računski stek
        INC     HL             Preskače se zarez i
        INC     HL             vrši pomeranje na
        INC     HL             početak drugog parametra (y)
        CALL    &33B4          I drugi parametar ide na stek
        RST     &28            Poziv kalkulatora
        DEFS    &0F            Sabiranje
        DEFB    &38            Kraj računanja
        RET                    Povratak. Na računskom steku je
                               pripremljen rezultat za funkciju USR.
```

Povezivanje bejzika sa kratkim mašinskim potprogramima, sasvim je izvesno kod „spektruma" nije najsrećnije rešeno. Nedostatak naredbe CALL, koja bi prenosila parametre u oba smera obeshrabruje čak i iskusnije programere, tako da se obično radi ili na čistom bejziku, ili na čistom mašincu. A ako se odlučimo na mešavinu, na raspolaganju će nam stajati jedino funkcija USR sa svojim skromnim mogućnostima, pri čemu se parametri moraju prenositi pomoću gomile nepreglednih PEEK-POKE rutina. O tome smo već pisali u više navrata, ali tema je dovoljno interesantna da joj posvetimo još malo pažnje.

adresa i dužina. Na nama je, dakle, da pre povratka iz mašinskog programa postavimo na računski stek adresu i dužinu rezultujuće uređenog niza. Sam program ćemo recimo smestiti na adresu 60000.

Svakako u bejziku treba definisati neku funkciju **FN f$(x$)**, ali je odmah jasno da neće proći nešto kao **10 DEF FN f$(x$)=USR 60000.** Ne može sa jedne strane znaka jednakosti stajati veličina čija je priroda znakovna a sa druge strane veličina čija je priroda numerička. Računar će, sa pravom prijaviti sintaksnu grešku, i mi ćemo morati da zadovoljimo formu recimo sa

10 DEF FN f$(x$)=STR$ USR 60000

Sada, međutim imamo nov problem. Ne samo što moramo da sprečimo računar u izvršenju STACK_BC već ne smemo dopustiti da izvrši operaciju **STR$, kada bude završio sa funkcijom USR.**

Naravno podatak o tome da iza **USR** sledi **STR$**, takođe se nalazi na mašinskom steku dok traje izvršenje našeg mašinskog programa. Međutim ta informacija je dosta duboko u gomili drugih podataka.

Nema potrebe da ovde sada ulazimo u sve detalje oko sadržaja mašinskog steka, mada to, uz knjigu „Spectrum ROM Disassembly", i sat-dva rada sa dobrim disasamblerom, ne predstavlja veći problem, a pogotovo ne toliki da i sami ne pokušate analizu. Mi ćemo dati samo uputstvo za prepravku steka, i taj recept treba ponoviti u svakom programu koji kao izlaznu veličinu ima znakovni niz.

Pre svega, sa steka treba skinuti poslednjih šest adresa, odnosno 12 bajtova. To se može uraditi sa šest uzastopnih **PDP** instrukcija, ali je jednostavnije

```
LD      HL 12
ADD     HL SP
LD      SP HL
```

Konačno, na stek se moraju postaviti sledeća tri podatka: &0000 kao oznaka da je završeno izračunavanje aritmetičkog izraza, zatim &106E kao oznaka da je obavljena operacija **STR$**, i u samom vrhu steka &2764, što je adresa rutine za pripremanje alfanumeričkog rezultata.

Nov raspored podataka na mašinskom steku zavaraće bejzik interpreter. Čim završi sa naredbama našeg mašinskog programa „spektrum" će preći na adresu &2764 u okviru SCANNING. Stanje na mašinskom steku biće kao da je upravo završeno izračunavanje funkcije **STR$**, i kao da je odgovarajući rezultat smešten na računskom steku. A, u stvari, na računskom steku će se nalaziti naš podmetnut rezultujući niz, i on će biti dodeljen funkciji **FN f$.**

Primer 2

```
        ORG     60000
SORT_$  LD      HL,12           Sa mašinskog steka se
        ADD     HL,SP           uklanja šest poslednjih
        LD      SP HL           adresa
        LD      HL &0000        Zatim se
        PUSH    HL              na stek dovode
        LD      HL &106E        tri nova podatka
        PUSH    HL              za uspešan
        LD      HL,&2764        povratak u
        PUSH    HL              bejzik
        LD      IX (&5C0B)      Uzima se adresa parametara iz DEFADD
        LD      E (IX+4)        Početna adresa niza x$ ide
        LD      D,(IX+5)        u registar DE
        LD      C.(IX+6)        a dužina niza
        LD      B (IX+7)        u registar BC
        PUSH    BC              Sačuvaj dužinu niza za kasnije
        PUSH    DE              Sačuvaj početnu adresu niza
        RST     &30             Otvaranje radnog prostora dužine BC
        POP     HL              Uzmi adresu niza x$
        PUSH    DE              Sačuvaj adresu rezultujućeg niza y$
        LDIR                    Kopiraj niz x$ na mesto rezultata y$
        POP     DE              Obnovi adresu rezultata
        POP     BC              Obnovi dužinu
        CALL    &2AB1           Pozivom STK_STORE rezultat se smešta
                                na računski stek
        LD      HL &0001        Proveri da li je
        XOR     A               dužina niza možda
        SBC     HL BC           nula ili jedan?
        RET     NC              Vrati se ako jeste. Nema uređenja
        EX      DE HL           Početak niza prebaci u HL
        DEC     BC              Pripremi brojač dužine
SORT    LD      E &00           Signal: u nizu nije izvršena korekcija
        PUSH    HL              Sačuvaj početak i
        PUSH    BC              dužinu niza
PASS    LD      A (HL)          Uzmi element niza
        INC     HL              Pomeri se na sledeći
        CP      (HL)            Uporedi elemente
        JR      NC ORD_OK       Skok ako je redosled ispravan
        LD      D (HL)          U protivnom, vrši se zamena mesta
        LD      (HL) A
        DEC     HL
        LD      (HL) D
        INC     HL
        LD      E &01           Signal: izvršena je izmena u nizu
ORD OK  DEC     BC              Umanji brojač za jedan
        LD      A,B             Da li je to bio
        OR      C               poslednji par elemenata u nizu?
        JR      NZ PASS         Ako nije, idi nazad u petlju
        POP     BC              Obnovi brojač
        POP     HL              Obnovi adresu niza
        DEC     E               Da li je bilo izmena u nizu?
        JR      Z SORT          Ako jeste, idi natrag u petlju
        RET                     Povratak. Niz je uređen
```

Kompletan mašinski program koji simulira funkciju za uređenje niza dajemo u primeru 2. Odmah posle prepravke mašinskog steka, prikupljaju se podaci o nizu x$, on se kopira u radni prostor računara i tu se vrši njegovo sortiranje. Početna adresa i dužina rezultujućeg niza postavljeni su na računski stek i spremno čekaju povratak u bejzik.

Daćemo na kraju i jednu ideju kako se primenom alfanumeričkih nizova mogu ostvariti aritmetičke i druge operacije sa proizvoljnom tačnošću. Brojevi se mogu zadati u vidu nizova npr. „727", „17 935" itd. a mašinski program bi obrađivao takve brojeve i vraćao rezultat u vidu novog niza pri čemu ne bi bilo izgubljeno ni jedno cifarsko mesto. Savetujemo vam da bar pokušate sa programom za sabiranje. To će vam sigurno, dati podstreka i za ostalo.

Jovan Skuljan

automatski start

Pri pisanju bejzik interpretera za „komodor" očigledno se razmišljalo o potrebi samostartovanja programa po učitavanju. U interpreteru su stvari izvedene gotovo do samog kraja ali se jednim skokom na radni početak bejzika (warm start) sve prekida i računar ispisuje READY pa je RUN neophodno. Štete jer zaista nije trebalo dopisivati mnogo da bi se bar dobio oblik naredbe RUN „naziv programa" poznat na drugim računarima. Uz put je ubečena mogućnost da se više programa povezuje u lanac putem LOAD naredbe ali na način koji malo koga može da zadovolji.

Zbrka oko LOAD

Mehanizam početka izvršavanja bejzik programa na „komodoru" je dosta jednostavan, ali da bi se shvatio treba poznavati dve stvari — način pakovanja programskih linija i način uzimanja karaktera pri interpretaciji programa. Linije su složene od adrese na koju pokazuje sistemska varijabla TXTTAB (43 44 najčešće je to 2049) i to u formatu: dva bajta koja predstavljaju apsolutnu adresu sledeće programske linije, dva bajta koja sadrže linijski broj i zatim sadržaj linije sa tokanizovanim naredbama i ostalim podacima u običnom ASCII kodu, te na samom kraju jedan bajt nula. Kada na mestu drugog bajta (viši bajt adrese sledeće linije) piše 0 onda je to oznaka kraja celog programa. Uobičajeno je da u tom slučaju i prvi ima istu vrednost pa za kraj programa stoji oznaka 0,0,0. Čuvanje apsolutne adrese početka sledeće linije, a ne dužine linije doprinosi naznatno brzini izvršavanja programa, ali postavlja i dodatni zahtev izmene ovih adresa pri svakoj promeni programa ili kasnijem upisivanju u računar od neke druge adrese u odnosu na onu na kojoj je pisan.

„Komodor" nakon LOAD ne proverava da li je program pisan od iste adrese na koju se sad učitava, već jednostavno uvek obavlja sređivanje ovih adresa. Sam postupak je poveran kratkoj rutini u bejzik ROM-u na adresi 42291 i odvija se tako što se ispituje bajt po bajt i traži nula na kraju linije pa se na osnovu njene adrese postavlja vrednost pokazivače sledeće itd. Treba napomanuti da rutina ne postavlja vrednost sistemske varijable VARTAB (45 46), kraj programa i početak bejzik varijabli već samo prestaje kada na njega naiđe. U postupku se koristi indirektno indeksirano adresiranje, konkretno LDA (34), Y gde se Y registar koristi kao preskok za bajt koji se ispituje, pa je neophodno da se nula nađe uvek u sledećih 255 bajtova. Ovaj uslov je za bejzik programe uvek ispunjen jer je dužina linija ograničena ekranskim editorom na 80 znakova, ali ako se postupak greškom primeni na mašinski program postoji velika verovatnoća da će rutina ne našavši nulu neprestano da „vrti" sadržaj Y registra u krug i tako blokira računar. Još gori slučaj nastaje ako se u tom opsegu i pojavi nula. Rutina će tada na osnovu njene adrese da postavi vrednost prva dva bajta od kojih je počela ispitivanje i tako promeni mašinski program pa je nakon SYS pravi krah neizbežan.

Uzbuna, RUN!

Kako započinje izvršavanje programa posle RUN? Ako odbacimo uobičajene priprame (RESTORE CLR itd) koje mogu i da se preskoče ostaje samo jedna operacija: sistemskoj varijabli na adresi 122 123 treba dodeliti vrednost apsolutne adrese umanjene za jedan, od koje želimo izvršavanje programa. Jasno, na ovoj adresi se mora nalaziti kod neke naredbe ili bajt 0 kao kraj linije. Kada je adresa postavljena, obavlja se JMP (776) (vektor IGONE za rutinu BCD,

```
PRIMER 1-

1 REM  AUTO-START LOADER
2 REM
3 REM                          PROFFI-ASS 64
4 REM
5 OPEN8-8 2 'LOADER.P W':REM OTVARANJE
6 REM                          DATOTEKE ZA
7 REM                          PROGRAM
10 SYS28672             ;POZIV ASEMBLERA
20  OPT O2              ;KOD U DATOTEKU
30 *= 256               ;ORG OD ADRESE 200
40 COUNT = 258          ;
50  BYT 2               ;258 PUTA BAJT 2
60 COUNT = COUNT - 1    ;
70 -IF COUNT!-GOTO 50;
80 NOP:NOP:NOP          ;
88 ;                    ;OVAJ DEO CE BITI
81 ;                    ;NA ADRESI 515
100 LDA #100            ;LOGICKI BROJ
110 LDX #8              ;BROJ DISKA
120 LDY #1              ;SEKUNDARNA ADRESA
130 JSR 65466           ;POSTAVI LA FA,SA
140 LDA #(KRAJ-NAME)    ;DUZINA NAZIVA
150 LDX #<NAME          ;
160 LDY #>NAME          ;POCETNA ADRESA
170 JSR 65469           ;POSTAVI NAZIV
180 LDA #0              ;
190 JSR 65493           ;LOAD
200 STX 45:STY 46       ;KRAJ PROGRAMA U
201 ;                   ;VARIJABLU VARTAB
210 JSR 42585           ;CLR RESTORE
220 JSR 42291           ;SREDI LINK ADRESE
230 JMP 42926           ;RUN
240 NAME  ASC "NAZIV GLAVNOG PROGRAMA"
250 KRAJ  END
READY
```

upućivanja na izvršenje naredbi) i program započinje se radom od zadatog mesta.

Sve ovo se reflaktovalo na naredbu LOAD tako što su razdvojena dva slučaja: (a) program se upisuje iz direktnog moda i (b) program se upisuje iz drugog programa koji je u toku. Prvi slučaj je lako realizovati. Kada se program učita, poslednja adresa koju je zauzeo dodeljuje se varijabli VARTAB, sređe se adrese za vezu sa sledećom linijom, izvede RESTORE i CLR i ono što je interesantno postavi adrese u lokacijama 122 123 na prvi bajt programa kao da će započeti RUN ali odmah sledi skok u direktni mod čime se ovo menja.

U slučaju (b) umesto skoka u direktni mod izvodi se JMP(776) čime automatski počinje izvršavanje novoupisanog programa. Izbačen je CLR i dodela vrednosti varijabli VARTAB se namerom da bejzik varijable iz prethodnog programa budu sačuvana, jer jedino time povezivanje programa na ovaj način dobija smisao. Sve je to lepo zamišljeno ali postavlja ograničenja, drugi program ne može biti duži od prvog jer VARTAB ostaje isti pa varijable dobijaju besmislene sadržaje koji su, u stvari bajtovi upravo upisanog bejzik programa a ujedno promena njihovih vrednosti menja program. Sve u svemu prilično lepa zabava kada nakon verovatnog SYNTAX ERROR zadate LIST. Iako ovo ograničenje i nije tako ozbiljno, navedena tehnika se na „komodoru" retko koristi.

Konstruktori „komodora" su uglavnom izabrali tri načina rešenja problema. Prvi samo delimično rešava pitanja upisivanja i startovanja mašinskih programa, drugi se primenjuje za bejzik dok je treći „pravi" i u komercijalnim programima predstavlja gotovo obaveznu tehniku.

Jednostavno i — kuso

Kada napišete mašinski program, možete ga za korišćenje „spakovati" na dva načina. Napišete bejzik liniju na primer 10 SYS 2061 a onda mašinski program smestite od adrese 2061, koja je iza tri nule koje zatvaraju bejzik program (u ovom slučaju samo jedna linija). VARTAB se pre snimanja promeni se POKE da pokazuje na kraj mašinskog programa. Kada se program kasnije učita sređivanje adresa na početku linija neće preći u područje mašinskog programa jer će prethodno biti zaustavljeno sa navedena 3 bajta 0.

Drugi način daje veće mogućnosti i podrazumeva poseban bejzik program koji će upisati mašinski (ili više njegovih delova) i obično izgleda ovako:

```
10 IF C=O THEN C=1. LOAD "naziv
   prvog dela" 8 1
20 IF C=1 THEN C=2  LOAD ,naziv
   drugog dela" 8 1
30 IF C=2 THEN C=3
100 SYS pozivna adresa
```

Primećujete u LOAD naredbi sekundarnu adresu 1 koja znači da program treba upisati na izvornu adresu. Ovo je obavezno jer se podrazumeva da mašinski program u ovom slučaju ne zauzima početne adrese bejzik područja. Nakon LOAD pošto se radi o programskom modu varijable ostaju očuvane i program ponovo startuje iz početka ali sada varijabla C ima vrednost 1 pa počinje upisivanje drugog dela itd sve dok se ne stigne do SYS naredbe.

Mogućnost startovanja programa na „Komodoru" odmah nakon upisivanja sa spoljne memorije „zvanično ne postoji — ako se ne računa taster RUN za upis i startovanje programa sa kasetofona — ali je auto-start ipak izvodljiv zahvaljujući pukoj slučajnosti — mašinski stek i INPUT bafer u prostoru sistemskih varijabli zauzimaju međusobno pogodan položaj. Malo poznavanja mašinskog jezika i rutina LOAD operativnog sistema daje mogućnost ne samo da se zaboravi na naredbu RUN nego i da se programi zaštite od presnimavanja   dobro, bar toliko da „pirati" imaju čime da se zabavljaju.

Ograničenje vezano za razliku u dužini programa može se prevazići simulacijom upisivanja iz direktnog moda. Program koji treba da obavi upisivanje drugog jednostavno prestaje sa radom ali pre toga u red za tastaturu postavlja sekvencu LOAD „naziv", CHR$(13) RUN CHR$(13). Računar prelazi u direktan mod a odatle nadalje sve je isto kao da ste sami otkucali navedene znake. Ovo je komodor primenio na taster RUN. Ako ste prebrojali slova koja u ovom slučaju treba smestiti u red za tastaturu videćete da tražimo nemoguće: bez naziva programa ima ih preko 10 što je maksimalan broj koji u red može da stane. Za prostije slučajeve naredbe se mogu upisati u skraćenom obliku slovo L pa zatim SHIFT+O. Međutim ekranski editor pruža veće mogućnosti. Obrišite ekran i ispišite potrebne naredbe (sve iz programa) a u red za tastaturu postavite samo kodove HOME (CHR$(19)) i onoliko CHR$(13) koliko linija na ekranu treba da se izvede. Kada program prestane sa radom desiće se isto kao da ste sami vratili kursor na neku liniju i pritisnuli RETURN. Treba pažljivo isplanirati gde ispisati potrebne naredbe. Ako prva linija ima naredbu LOAD „naziv programa" kada se izvrši kursor se neće naći jednu liniju ispod nje već četiri (!) jer je računar ispisao i READY pa RUN mora biti na petoj liniji da bi ga sledeći CHR$(13) iz reda izveo.

Svi ovi načini ma koliko korisni i interesantni bili zapravo ne rešavaju osnovni problem i dalje je potreban bar jedan RUN a i zaštite koji bi se u okviru njih izvodile bile bi trivijalne. Pravo rešenje se zove mašinski stek.

Trikovi sa stekom

Očigledno bez poznavanja mašinskog jezika „komodor" ne može da pokaže sva svoje kvalitete. Kada bejzik naredba LOAD obavi pripremne operacije poziva potprogram operativnog sistema sa istim nazivom očekujući da se po obavljenom poslu mikroprocesor ponovo vrati na nju da bi se obavio ostatak sređivanja adresa za vezu sa sledećom linijom CLR itd. Ako se to i desi propada svaka mogućnost da se umešamo i uradimo nešto drugo, ili isto sa dodatkom RUN. Rešenje je da, umesto u uobičajena memorijska područja počnemo upisivanje od adrese 256 dakle preko samog steka i tako na njega postavimo drugu povratnu adresu nakon LOAD operativnog sistema. U prvi mah ovo izgleda komplikovano jer se ne zna gde se u tom trenutku ova adresa nalazi, odnosno koja je vrednost registra SP (stek pointer). Zahvaljujući činjenici da je upisivanje u memoriju izvedeno tako da između postavljanja jednog primljenog bajta u memoriju i nastavka procesa nema RTS instrukcija, može se ceo stek izmeniti. Samom procesu upisivanja ovo zbog toga neće smetati. Iako će rutina

```
PRIMER 2

10 REM  UNIVERZALNI AUTO-START LOADER
11 !
12 !
20 OPEN 2,8,2,"NAZIV LOADERA,P,W"
30 PRINT#2,CHR$(0)CHR$(1);
40 FORI=1TO255:PRINT#2,CHR$(2);:NEXT
50 RESTORE
60 FORI=1TO36:READB:PRINT#2,CHR$(B);:NEXT
70 READB$:PRINT#2,B$:CLOSE2
80 DATA169,188,162,8,160,1,32,186
81 DATA255,169,000,162,38,160,2,32
82 REM UPISATI 11 DUZINU NAZIVA PROGRAMA
83 DATA189,258,169,0,32,213,255,134
84 DATA 45,132,46,32,89,166,32,51
85 DATA165,78,177,167
88 DATA"NAZIV PROGRAMA"
READY
```

prebrisati deo naših izmena (pošto i u okviru nje poziva druge subrutine) ipak će, kao i svaka druga koja treba da radi „pošteno" sve što na stek stavi sa njega i da skine i tako dođe do prave povratne adrese koju neće izmeniti. Samim tim naša izmena ostaje nedirnuta i poslednji RTS ne vraća mikroprocesor u bejzik odakle je došao.

Mali problem predstavlja činjenica da je adresa dvobajtna a vrlo je teško predvideti da li će vrednost SP biti parna ili neparna. Zbog toga je jedino sigurno rešenje postaviti povratnu adresu čiji su viši i niži bajt isti na primer (4 4=1028). Prva ovakva adresa iznad steka je 514 (2,2) a to je područje INPUT bafera dakle u toku LOAD slobodnih 89 bajtova! Treba još imati u vidu da mašinska RTS instrukcija povratnu adresu sa steka pre povratka poveća za jedan pa tako program mora počinjati od adrese 515.

U primeru jedan je dat asemblerski listing programa koji će se sam startovati upisati glavni program i startovati njega. Pošto program treba asemblirati na adresu 256 korišćena je osobina Proffi-ass 64 da rezultujući kod smesti direktno u programsku datoteku i pri tome prva dva bajta budu adresa od koje će se kasnije učitati. Korišćena je i mogućnost kreiranja petlji da bi se izbeglo pravljenje tablice od 256 bajtova 2.

U obliku u kome je dat program omogućuje upisivanje i startovanje bejzik programa. Za ovo su odgovorne poslednje tri instrukcije ali je za mašinske programe stvar još jednostavnija umesto njih staviti instrukciju JMP sa pozivnom adresom.

Zaštita bez zaštite

Pre navedene tri instrukcije može se dopisati deo koji će sprečiti dejstvo STOP i RESTORE tastera. Program tada neće moći biti zaustavljen radi kopiranja, ali se ne radi ni o kakvoj zaštiti glavni program je i dalje odvojen, pa se lepo može upisati i bez „loadera"! Slabije rešenje je da se „loader" snimi zajedno sa glavnim programom sa svim bajtovima između i tako napravi jedna celina. Bolje je izvesti sledeće: pre snimanja glavnog programa nad svakim njegovim bajtom izvedete XOR operaciju sa nekim brojem. Sam program tako gubi smisao dok se identična operacije ne ponovi, a ovo će biti zadatak „loadera". Stvar se može komplikovati i rotiranjem bitova i dodavanjem neke konstante, ili pravljenjem čitavih tablica za dešifrovanja.

Sve ovo lepo radi ali je nemoćno pred raznim programima za kopiranja. Ako šifru ne upišete u program već je „loader" zatraži pre svakog učitavanja, program će moći da se kopira ali niko sem vas neće moći da ga koristi što je za ličnu upotrebu gotovo 100% zaštita. Proizvođači ovaj metod ne mogu da koriste, pa se dovijaju uglavnom na jedan način. Na disku se na nekom sektoru namerno napravi greška. „Loader" pre upisivanja proverava da li takva greška postoji, što bi trebalo da bude pouzdana indikacija da disketa nije kopirana. Grešaka ima raznih — neke se mogu i prekopirati ali određen broj teže. Precision Softwera (autori čuvenog EASY SCRIPTA) svoj program EASY SPELL su zaštitili tako da prvi „Loader" upisuje drugi dešifruje ga upisuje sledeći   i tako ravno pet puta! Tek poslednji upisuje i dešifruje glavni program, a tek u njemu (a dugačak je oko 12K) se vrši provera da li je disketa kopirana. Nije pomoglo! Očigledno da zaštita programa leži u nekim drugim sferama ljudskih delatnosti.

U primeru 2 dajemo program koji će kreirati „loader" za bilo koji vaš program. Upišite nazive koje smo ostavili otvorenim i u liniji 81 dužinu naziva glavnog programa. Nakon RUN na disku će se naći program čijim učitavanjem automatski započinje upisivanje i izvršavanje glavnog programa.

Zoran Živošić



— Pođi sa mnom u moj stan dušo .. da ti pokažem na licu mesta moju tehniku interfejsa!

simulanti i modeli

Simulacije

Ideja metoda je poznata već vekovima ali je postala aktuelna tek sa pojavom računara. Uopšte, računari su inspirisali nastanak ili ponovno bavljenje mnogim naučnim disciplinama: primenjena statistika, ekonometrija, matematičko programiranje (čiji je najpoznatiji slučaj linearno programiranje), zatim novi metodi u numeričkoj matematici, operaciona istraživanja i slično. Kada su čisto algebarski modeli nepodesni za primenu, direktno imitiranje proučavanog sistema ostaje jedina mogućnost. Takvi sistemi sami po sebi su ogromni, pa i simulacioni programi kojima ispitujemo takve sisteme zahtevaju znatan utrošak ljudskog i mašinskog vremena. Na primer, operacije u nekoj luci su isuviše kompleksne da bi se mogle prikazati preko jedne ili više formula. Takav sistem moramo simulirati na računaru. Da bi smo odredili mesto računarske simulacije u procesu istraživanja, moramo naučiti šta su

Modeliranje sistema

Sistem je skup delova funkcionalno organizovanih u povezanu celinu. Ovo je zaista, opšta definicija, pod koju bi se mogli podvesti i telefonski sistemi, računarski sistemi, sistemi sa čekanjem itd. Za sva sistema postoje zajedničke karakteristika

- antitet: objekat od interesa u sistemu,
- atribut: svojstvo antiteta,
- aktivnost: proces koji prouzrokuje promenu u sistemu,
- događaj: početak ili završetak aktivnosti, alternativno, odigravanja promene u nekom momentu

Uzmimo primer samoposluga. Šta bi bili antiteti u jednoj tako običnoj okolini? Da odgovorimo na to pitanje moramo znati zašto uopšte razmatrati samoposlugu na „naučnoj bazi"? Cilj proučavanja sistema je da se njegovo ponašanje promeni, tj. uskladi sa našim potrebama. Samoposluge su dobre — osim u jednom: dugo se čeka na izlasku. Što se tiče kupaca, idealno bi bilo da svakog kupca u svakom trenutku čeka prazno naplatno mesto. Ovo je, naravno, nemoguće. Uprava samoposluge ima drugi cilj: zaposliti ekonomičan broj službenika tako da promet bude što je moguće veći a rashodi na osoblje što je moguće manji. Najmanji rashod je kada radi samo jedna kasa na izlazu, ali bi tada red u iole većoj samoposluzi bio ogroman. Cilj je odrediti optimalan broj aktivnih kasa tokom dana. Sada je lako odrediti entitete u samoposluzi — to su kupci i kase. Svaki kupac a i svaka kasa, mogla bi imati ogroman broj atributa, no mi se odlučujemo samo za one koji su neposredno vezani za prolaznost pored kasa: broj artikala koje mušterija nosi, veština kasirke i slično. Primer loše odabranog atributa (u skladu sa našim ciljem u ovom primeru) može biti boja očiju kupca — ona nema veze sa procesom opsluživanja i povećanjem prolaznosti kroz naplatno mesto.

Slična razmatranja odnose se i na aktivnosti i događaje. Biranje artikla u samoposluzi ovde nije bitno, mada bi moglo biti od interesa u nekom drugom istraživanju. Događaj u našem sistemu je stupanje kupca u red čekanja, kao i napuštanje reda čekanja pošto je platio robu. Događaj i aktivnosti su usko povezani: trenutak u kome se mušterija priključuje redu čekanja svakako je događaj u našem sistemu.



Slika 1 Informatički pristup pojmu sistema

Cilj identifikovanja antiteta, atributa, aktivnosti i događaja je merenje frekvencija pojavljivanja, tj. prikupljanja podataka na organizovan način. Po završenom prikupljanju podataka, možemo računati razne statistike i na osnovu njih zaključivati o ponašanju sistema.

Umesto proučavanja sistema — proučavamo njegov model. Model je bilo koje predstavljanje sistema, ali je prirodno zahtevati još dve stvari:

- model treba da nam omogući predviđanje ponašanja sistema, i
- sa modelom treba da se lakše radi nego sa samim sistemom.

Teorija je opšta formulacija principa koji je izvučen iz sistema, odnosno iz merenih podataka sa sistema. Cilj postavljanja teorije je da se objasni ponašanje sistema i — eventualno — da se prognoziraju njegova buduća stanja. Model uvek predstavlja istovremeno i sistem i odgovarajuću teoriju. Drugim rečima, teorija objašnjava sam sistem, služi kao osnova za izgradnju modela, povezujući model i sistem. Jedina svrha modela je provera teorije u praksi. Naša veština biće u tome da napravimo model koji „hvata" strukturu sistema — ali tako da se sa njim lakše radi nego sa samim sistemom.

Slika 1 prikazuje „informatički" pristup sistemima: na date ulazne parametre sistem reaguje proizvodeći neke izlazne parametre. Slika 2 je situacija u kojoj umesto sistema koristimo model; to je ujedno, i opšta šema modeliranja. Suština je u zaobilaženju sistema — putanja A-B-C-D ima više koraka nego direktno E-F ali je lakša, „jeftinije" njome proći nego direktno kroz E-F.

Modeliranje sistema je vrlo opšti pristup i koristi se u raznim situacijama i u razne ciljeve. Najčešće cilj je predskazivanje ponašanja sistema. Često se koristi u projektovanju novih proizvoda, ali i ne mora: igra avantura je model za „sistem" koji nikad neće realno postojati. Sisteme modeliramo i u slučaju da se podaci ne mogu dobiti direktno, kao u slučaju svemirskog šatla. Sisteme modeliramo i kada treba poboljšati kontrolu nad nekim delom sistema, doneti neku odluku, promeniti strategiju itd.

Deterministički i stohastički modeli

Modele možemo razvrstavati na mnogo načina. U fizičke modele spadaju: globus, planetarijum, tunel vetrova, simulator leta itd., a nasuprot njima postoje apstraktni modeli: budžet, hemijska formula, programska blok-šema, matematička jednačina itd. Po sličnosti, fizički modeli dele se na ikonske i analogne. Ikonski model spolja izgleda identično pravom sistemu kojeg predstavlja, npr. kabina simulatora leta (samo je mnogo sigurnija i jeftinija). Analogni model se ponaše kao sistem ali je fizička realizacija drastično različita. Kretanje klatna pokorava se istoj diferencijalnoj jednačini kao i tok struje u elektronskom kolu. Masu klatna možemo izračunavati iz karakteristika struje u takvom kolu, koje su analogni model mehaničkog sistema. Svi apstraktni modeli mogu se izraziti i kao matematički modeli, tj. u obliku nekih matematičkih relacija. Matematički modeli su najvažniji, jer se mogu realizovati kao program za računar.

Sistemi mogu biti deterministički ili stohastički. Deterministička varijable predstavljaju poznata stanja sisteme, tj. stanja koja se mogu precizno meriti, odnosno izračunati. Stohastičke varijable označavaju parametre sistema koji nisu tačno poznati unapred — ali otprilike znamo kako će se ponašati. Njima obuhvatamo slučajne karakteristike u sistemu. Prilikom kreiranja modela, jedno od najvažnijih pitanja koja treba razrešiti je „da li je model deterministički ili stohastički?" Naime, jedan matematički aparat se primenjuje na deterministička modele (obično izvodi, integrali i/ili diferencijalne jednačine), dok se sasvim druge matematička discipline koriste u stohastičkim sistemima (verovatnoća, statistika, slučajni procesi). Analitičar (nazovimo tako osobu koja kreira model) mora u svakom posebnom slučaju da odluči o prirodi modela, ali je potrebno podvući činjenicu da prisustvo makar i jedne slučajne pro-

Simulacija na računaru je jedan od najsnažnijih i, po primenama, najraznovrsnijih metoda za rešavanje problema Danas se koristi u preko 70 naučnih oblasti, u koje spadaju planiranje rasporeda ljudstva projektovanja aviona, urbanizam, predviđanja berzanskih tokova, projektovanja pristaništa i luka fabričkih postrojenja, razvoj računarskih sistema, vojne operacija, organizacija rada bolnica, robnih kuća prodavnica itd Metod simulacija koristi se za analizu sistema koji su previše komplikovani da bi se analizirali običnim formulama, ili za koje čak nikakve formule i ne postoje.

menljive automatski pretvara deterministič-ki model u stohastički

Preciznije i sistem i model mogu biti bilo deterministički bilo stohastički pa postoje četiri moguće kombinacije

- deterministički sistem — deterministički model ne primer jednačina kretanja prostog klatna možemo je rešiti analitički ili numerički, u ovom slučaju računar se koristi samo kao super-brzi digitron,
- deterministički sistem — stohastički model, primer za ovo je računanje površine ispod neke krive pomoću metoda Monte-Karlo
- stohastički sistem — deterministički model, ovo je pomalo čudna kombinacija a primer je generisanje slučajnih brojeva pomoću računara u simulacijama diskretnih sistema (raskrsnice saobraćaj, sistemi sa redovima itd) od suštinske važnosti su algoritmi za generisanje slučajnih brojeva
- stohastički sistem — stohastički model ovaj slučaj redovno se razrešava metodom simulacije

Statički i dinamički modeli

Modeli se često dele po trajanju vremenskih segmenata u kojima se događaju sistemske aktivnosti U tzv statičkim modelima stanja sistema posmatraju se u jednoj vremenskoj jedinici dok se u tzv dinamičkim modelima stanja sistema posmatraju u različitim trenucima jer se neke promenljive menjaju tokom vremena

Jedna opšte podela po vremenu daje nam kontinualne i diskretne sisteme Pretpostavimo da u svakom modelu postoji interni sat koji počinje sa odbrojavanjem baš u trenutku u kome počinje naše posmatranje sistema Vreme u kojem posmatramo je dugačko T vremenskih jedinica pa je u pitanju interval (0 T) Slika 3 prikazuje model sa kontinualnim (gore) i diskretnim (dole) vremenom Grubo računo ako je kriva koja opisuje stanja sistema neprekidna — sistem je sa kontinualnim vremenom, inače je sa diskretnim vremenom Modeli sa diskretnim vremenom dele se na modele kritičnog događaja i modele sa konstantnim vremenskim priraštajem U modelu kritičnog događaja interni sat zavisi od događaja u modelu, te se posmatranja sistema vrše u slučajnim vremenskim jedinicama Primer za to je model saobraćajne raskrsnice gde se događaji (nailaseci vozila) odigravaju u slučajnim vremenskim trenucima Istu raskrsnicu možemo obrađivati i kao model sa konstantnim vremenskim priraštajem ako merimo broj vozila na raskrsnici recimo svakih 5 sekundi

Na kraju spomenimo i podelu koja se najčešće sreće u praksi na diskretne i kontinualne modele Diskretan model zadovoljava dva uslova (1) postoje (veliki) blo-



Trivijalan problem — izvanredni efekti kompjuterske simulacija sisteme za osvetljavanje ulice u jednom projektu Nacionalne istraživačke laboratorije za puteve u Velikoj Britaniji

kovi vremena u kojima se stanja sistema ne menja i (2) sve pojave sistema izražavaju se u celim brojevima Raznolike saobraćajne situacije zadovoljavaju ova dva uslova te se uvek i modeliraju kao diskretni sistemi Isto važi i za sisteme sa čekanjem, na koja se svodi veliki broj sistema koji su interesantni u praksi Samoposluga je klasičan primer sistema sa redovima čekanja

Najvažnija osobina diskretnog sistema je da „skača" iz stanja u stanje, te se zadaje kao niz stanja, uz akcija koja sistem prevode u sledeća stanja

Dve su osnovne karakteristika kontinualnih moela (1) stanja sistema menjaju se neprekidno u odnosu na vreme i (2) merne jedinice sistema nisu obavezno celobrojne Očit primer je putanja rakete u letu raketa menja položaj u svakom vremenskom trenutku i to neprekidno Matematički modeli kontiualnih sistema obično se zasnivaju na diferencijalnim jednačinama Npr prvi izvod je brzina rakete drugi izvod je ubrzanje i slično

Nije lako unapred odrediti koji tip modela najbolje zadovoljava u određenoj situaciji Kako, recimo modelirati proces rasta neke populacije (npr koliko ljudi će biti na svetu 2000-te godine?) Prvo što nam pade na pamet je diskretan model jer se rođenja odigravaju u diskretnim (i nepredvidljivim) momentima a i broj ljudi je uvek celobrojan Međutim vreme između uzastopnih umiranja i rađanja je malo — u odnosu na raspon događaja u sistemu (trajanje sistema dato je u godinama) Tada je bolje obrazovati diferencijalnu jednačinu koju ćemo takse rešiti numerički nego da u računaru organizujemo milione „umiranja i rađanja" koja zahteva pristup diskretnom simulacijom Oba pristupa su u ovom slučaju podjednako dobra (svaki sa izvesne tačke gledišta), pa druge okolnosti presuđuju Konkretno odlučili bismo se za kontinualnu sumutaciju, jer je vreme izvršavanja programa na računaru kraće

Jedino opšte pravilo je da sve zavisi od istraživačeve ličnosti predznanja, naučničkih naklonosti, raspoloživog vremena i mašina itd U istorijskom pogledu kontinualni sistemi dominiraju od 17-tog veka sve do pojave računara dakle, sve do pre nekih 30—40 godina

Rešavanje modela

Pošto smo se odlučili za model treba ga rešiti odnosno, izvući što je moguće više informacija o ponašanju sisteme na osnovu ponašanja modela Opšt postoje dve pristupa (1) analitički i (2) numerički Model je analitički ako opisuje sistem kroz matematička formule ili relacije Istraživači najradije barataju sa gotovim formulama jer pružaju utisak elegancija i snage Možemo ih računati ručno a u većini programskih jezika kodiranja formula je trivijalno Znači da je zadatak predviđanja ponašanja sistema čvrsto pod našom kontrolom — sve što treba uraditi je provesti niz lakih aritmetičkih operacija

Avaj nije sve tako ružičasto Modeliranja uvek znači i idealizaciju posmatrana pojave nezavisno od tipa modela Kod matematičkih modela proces apstrakcija doveden, je do ekstrema, pa sačinjeni model može davati vrlo grubu sliku o ponašanju sistema Dalje analitički izrazi mogu biti vrlo komplekani, a postoje i izrazi koji se ne mogu direktno izračunavati npr nelinearne parcijalne integro-diferencijalne jednačina Tada moramo pribegavati numeričkim metodama, ali to unosi grešku numeričkog metoda u celu simulaciju Za utehu, ostaje činjenica da se svaka formule može izračunati preko Tajlorovog reda sa bilo kojom traženom tačnošću — samo ako smo voljni da „platimo" odgovarajućim ljudskim i mašinskim vremenom

Postoji i dublja razlika između analitičkog i numeričkog pristupa rešavanju mode-

je. Analitički pristup nudi opštost (u okviru dobro definisanih ulaznih pretpostavki) i omogućava da se sistem analizira kvalitativno. Suprotno tome, numerički metodi nude samo delimičnu sliku ponašanja sistema, a ako insistiramo na poznavanju celokupna slika o sistemu, numerički proračun se mora ponavljati sve dok se sve kombinacije ulaznih podataka ne sračunaju. To je često nemoguće u praksi, te postoji opasnost da se zaludimo nepotpunim podacima — koja smo još mi sami napravili! U praksi, težnja je da se sistemi rešavaju analitički, a da samo neophodni delovi budu numerički.

Računarska simulacija

Analitička rešenja su prilično retka. Napraviti jedno novo analitičko rešenje svodi se na otkrivanje novih matematičkih teorija i mnogi istraživači nisu za to sposobni. Rezultati su skoro uvek potrebni „se, rokom juče", nema se vremena za fundamentalna istraživanja. Jedina mogućnost su numerički proračuni, ali ih je najčešće nemoguće izvoditi ručno: računari prirodno ulaze u igru u ovom trenutku. Na žalost, tačno definisati računarsku simulaciju prilično je teško. U nekom najopštijem smislu simulacija je metod izvođenja eksperimenta na modelu sistema, iako moguća, simulacija je danas nezamisliva bez računara. „Hakerska definicija" bi bila da je simulacija veština pisanja i igranja sa računarskim modelom. Ovo je gruba definicija, jer bi tada bukvalno svaki program bio simulacija nečega (ako ništa drugo, ono bar mentalnih procesa u programerovoj glavi), što je sasvim nepraktično. Ako se ograničimo na oblast diskretnih i stohastičkih simulacija, tada je dobra sledeća definicija: „Simulacija je eksperiment pomoću apstraktnog modela sistema koji je realizovan kroz računarski program u kome svi procesi traju izvestan vremenski period". Dakle, sam izraz „računarska simulacija" obaveštava nas da se simulacija izvršava na nekom računaru, a tip simulacije se ne objašnjava pobliže.

Osnovna ideja u modeliranju je da istraživač eksperimentiše nad modelom (dakle simulira sistem), jer se to ne može uraditi sa stvarnim sistemom. U svakom slučaju računarski model ima nekih bitnih prednosti nad ostalim vrstama modela. Programer uvek kontroliše sve uslove eksperimente koji se može ponavljati sve dok je potrebno (ovo je često neizvodljivo u realnosti!). U velikom broju slučajeva istraživač se prihvata simulacije kao sredstva za upoznavanje nekog interesantnog sistema. Jedan stari programerski sarkazam kaže da tek pošto programer završi program uvida kako je zapravo trebalo da ga piše od samog početka! Ovo je naročito tačno i važno u simulacijama sistema koji još ne postoje, a fizički modeli se ne mogu konstruisati (ljudska kolonija na Mesecu). Uopšte, napredujući preko raznih verzija, istraživač prikuplja dragocene podatke o sistemu kojeg proučava, što je jedna od fundamentalnih prednosti simulacije nad svim ostalim istraživačkim metodama.

Verifikacija i valjanost

Pošto smo odabrali teoriju o ponašanju sistema i odlučili se za tip modela, kao i za oblik realizacije [illegible] pr[illegible]blema

• verifikacija [illegible] da [illegible] stavlja teoriju, odnosno da [illegible] ispravno prevedena u model, kao [illegible] ispitivanja valjanosti m[illegible], dakle da li se model ponaša ekvivalentno [illegible]

Problem verifikacije poz[illegible] ramiranju kao problem korektnosti: da li program radi ono što [illegible] radi? [illegible] sada ovaj problem ostaje [illegible] koje, osim za najjednostavnije [illegible] ne možemo tvrditi da su korektni. [illegible] rečeno, verifikacija je osnov[illegible] gramera, dok je analitičar v[illegible] zainteresovan za proveru valjan[illegible] njemu i teže valjanost s[illegible] tak



Slika 2 opšta šema modeliranja

Slika 3 Kontinualni i diskretni sistemi

pošto je program verifikovan, pa ako model ipak ne imitira sistem dovoljno dobro, onda nije jasno da li je greška u loše napisanom programu ili u loše postavljenom modelu (ili u teoriji po kojoj je pravljen program itd.). Da bi se eliminisao ovaj razorni crv sumnje, analitičar postaje i programer: svi pravi analitičari kad-tad nauče neki jezik programiranja. Kako njih ipak interesuje više da dobiju rezultate o svom sistemu nego da provode dane i dane ryući se protiv računara, napravljeni su posebni simulacioni jezici.

To nipošto ne znači da se simulacije ne mogu praviti na bejziku, fortranu, paskalu itd. Naprotiv, oko 75% svih simulacija napravljeno je na fortranu! Međutim, model sistema na specijalizovanom jeziku piše se za jedan dan — [illegible] na fortranu za nekoliko dana do nekoliko nedelja, zavisno od spoljnih okolnosti. Otkuda tolika razlika u produktivnosti? Po nekim psihološkim merenjima, jedan programer ne može da napiše više od 30 naredbi BEZ OBZIRA na korišćeni jezik! A šta to znači, pogledajmo na primeru. Za jedan radni dan na mašinskom jeziku možemo napisati sabiranje dva broja u pokretnom zarezu; na paskalu ili fortranu može se napisati kompletn[illegible] procedura za numeričku integraciju, a na spe[illegible]ijalizovanom simulacionom jeziku kao što je D[illegible] imali bismo kompletan program za računarsku simulaciju njihanja klatna. [illegible] Razlika u produktivnosti je više nego u[illegible]ledna!

Da bi jedan specijalizovani jezik mogao da odgovori svojoj nameni, u njemu se mora nalaziti skoro sve što korisniku treba. Npr. simulacioni jezik za [illegible]iskretne sisteme mora da zadovolji sledeće [illegible]ove:

• Vreme. Analitičar može da [illegible]vede za s[illegible]e sistem, može da snimi simulaciono v[illegible]e, da promeni simulacioni sat itd.

• [illegible]ogađaji. Stanja modela treba da se [illegible]anja samo kada se dogodi nešto u sistem[illegible]. Ako se u istom trenutku odvija više [illegible]ocesa, mora se dati mogućnost analitičaru da [illegible]dabere strategiju prioriteta.

• Generator slučajnih brojeva. Mora biti zaista brz i kvalitetan.

• Obračunavanje važnih statistika. Treba da bude u[illegible]glavnom automatsko, a stand[illegible]an [illegible]bli[illegible] štampanog dokumenta da bi[illegible]e n[illegible] rasp[illegible]aganju analitičaru.

• Kontrola stanja simulacionog modela. Analit[illegible]ari treba pružiti mogućnost da sam [illegible] sva spoljašnje i unutrašnje komponent[illegible] modela.

• [illegible]ši zahtevi. Čitljivost programa, početna postavljanja, mogućnosti testiranja i otkrivanja grešaka, modularnost instrukcija i programskih struktura itd.

Metod simulacija koristi se u tri slučaja:

• istraživač želi da reši problem za koji postojeća matematička rešenja nisu adekvat[illegible] ili čak uopšte i ne postoje

• istraživač želi da nauči više o funkcionisanju sistema

• [illegible]stražuju se sistemi koji fizički ne postoje

Simulacija nije prosto kodiranje programa za računar. U širem smislu, metod s[illegible]lacija obuhvata veoma raznovrsne aktivn[illegible]sti kao što su:

• snimanje podataka na realnom sistem[illegible]

• eksperimentisanje sa realnim sistemom

• f[illegible]rmul[illegible]sanje teorija

• f[illegible]rmi[illegible]anja modela (jednog, a po potrebi [illegible]vi[illegible]e)

• p[illegible]sanje pr[illegible]grama za računar

• [illegible]ranje e[illegible]sperimenata sa računar[illegible] mod[illegible]m

• verifikacija dobijenih rezultata

• a[illegible]ali[illegible]a [illegible]nterpret[illegible]ranja i (eventualno) k[illegible]ćenje s[illegible]lacion[illegible]h rezultata

M[illegible]o je bolje simulirati ratne igre nego [illegible] sp[illegible]vodili u realnosti, kao što je bolje [illegible]mulirat[illegible] rad nove fabrike pa je tek [illegible]a sagradit[illegible]. Simulacija je opravdano [illegible]a jedan od najpopularnijih naučnih [illegible] d[illegible]ako [illegible] izmišljeni računari.

Dusko Savić

Radionica logičkih igara

poslednji uvek dobija

U „Računarima 2" i „Računarima 3" smo objavili škola akcionih i avanturističkih igara koja su, nadamo se pomogli domaćim vlasnicima računara da se uhvate u koštac sa na tako jednostavnim problemom programiranja igara U ovim i sledećim „Računarima" ćemo čitavu stvar dopuniti — videćemo kako se pišu logička igre U školi igara avantura, nismo, jasno, pisali program ranga „Hobbita", pa ni sada nećemo pisati program za šah Ipak, praćenje ovog napisa treba da pruži uvid u osnove tzv veštačka inteligencija da vam, ako ništa drugo, bar pruži priliku da razumete kako otprilike rade programi koji igraju šah i slična igre Pod pojmom „slična" podrazumavamo igre dva protivnika koji imaju podjednaka šansa za pobedu na tabli koja nije skrivena (igre tipa „Podmornica" ćemo programirati nekom drugom prilikom) i u kojima sreća tipa „slučajan broj" nema nikakvu ulogu

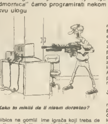
U toku ove Škole realizovaćemo nekoliko logičkih igara i to koristeći paskal Zašto baš njega? Pre svega, za pisanje logičkih igara su nam fundamentalno važne rekurzije koje se na većini bejzika mučno realizuju — programiranja rekurzija bi nas odvelo od osnovnog problema koji rešavamo Osim toga, bejzik je slabo standardizovan jezik dok će programi na paskalu raditi praktično na svim kućnim računarima Najzad, odustaćemo od korišćenja svih „specijalnih efekata" koje paskal nudi i tako učiniti programe dobrim zamenama za algoritamske blok dijagrame

Palidrvca bez vatre

Osnovni problemi koje treba da rešimo pre nego što počnemo da pišemo program koji će naučiti računar nekoj logičkoj igri su predstavljanje pozicije u memoriji računara, prikazivanje te pozicije na ekranu i izbor najboljeg mogućeg poteza u nekoj poziciji Drugi problem (prikazivanje na ekranu) je čisto tehničke prirode, pa se njime nećemo ozbiljnije baviti Prvi problem je takođe unekoliko tehničke prirode ali njegovo optimalno rešenje može da predstavlja neophodan uslov za rešenje trećeg suštinskog problema izbor poteza koji će kompjuter odigrati

Izbor najboljeg mogućeg poteza obično podrazumeva ispitivanje daljeg toka igre, tj poteze koje će protivnik odigrati u odgovoru na potez koji upravo razmatramo Da bismo u praksi ilustrovali ovu tvrdnju upoznaćemo jednostavnu igru koju bismo mogli da nazovemo „Poslednji dobija?" Na jednoj gomili je naslagano nekoliko palidrvaca Dva igrača naizmenično sa gomilice uzimaju jedno, dva ili tri palidrvca pri čemu se za svako uzeto palidrvce dobija po poen Igrač koji odnese poslednje palidrvca dobija dodatna dva poena Pošto ste upoznali pravila verovatno vam je jasan smisao upitnika koji smo stavili na kraj imena igre u trivijalnoj verziji igrač koji odnese poslednju šibicu obavezno dobija partiju dok je u verziji koju razmatramo dobija onaj ko ima više poena

Da bi se kompletno opisale pozicije do koja se došlo u ovoj igri potrebno je relativno malo podataka broj preostalih



Kako to misliš da ti nisam dorastao?

šibica na gomili ime igrača koji treba da odigra sledeći potez (A ili B) i naravno, trenutni broj poena igrača A i B Oznaka *4A21* na primer, označava da su na gomili četiri šibice da sledeći potez ima igrač A čiji je skor 2 dok igrač B ima samo 1 poen Jedan od mogućih daljih tokova igre je prikazan na slici1

```
slika 1:

                         7A00
A uzima 2                5B20
B uzima 3                2A23
A uzima 2                0B43
                         kraj
```

slika 2:



Suvišno je reći da u većini pozicija igrač može da izabere jedan od nekoliko poteza, što znači da, pri razmatranju pozicije, ne smemo da posmatramo samo jedan pravac daljeg toka partije Zato je na slici 2 prikazano takozvano stablo igre koja počinje sa pet šibica na gomilici i igračem A na potezu Svaka od grana ovoga stabla se završava nekom pozicijom ispod koje je ispisan skor posmatran sa aspekta igrača A +1 označava da je A pobedio sa jednim poenom razlike a −3 da je A imao tri poena manje od pobednika B

Stablo igre

Nema mnogo koristi od pisanja o stablima igre ako nismo u stanju da napišemo program koji će ta stabla generisati Zato ćemo pokušati da napišemo proceduru koja za zadatu poziciju, ispisuje stablo toka igre „Poslednji pobeđuje?", videćemo da će se umerenom modifikacijom ove procedure dobiti program koji redovno pobeđuje u ovoj igri Algoritam po kome radi procedura sa slika 3 bi mogao da bude

1 Ispiši poziciju P

2 Izračunaj koliko različitih poteza može da odigra igrač koji je zatekao poziciju P (ovaj broj može da bude i 0)

3 FOR svaki raspoloživ potez

4 Pripremi poziciju koja bi nastala odigravanjem poteza

5 Generiši stablo ove nove pozicije

6 NEXT potez

Ovakav algoritam posebno njegov korak 4 implicira korišćenje rekurzija procedura će očito pozivati samu sebe Da bi iz tog lanca pozivanja bilo izlaza u slučajevi-

ma kada je broj raspoloživih poteza nula biće ispisan krajnji skor

slika 3

```
[illegible]
```

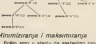
Ako otkucate, prevedete i izvršite program sa slike 3, dobićete rezultate sa slike 4 na kojoj je prikazana i njihova grafička reprezentacija (ukoliko vas zanima grafika pokušajte da sastavite program koji bi, za zadatu poziciju, zaista iscrtao drugi deo slike 4) Boljem razumevanju rada programa može da pomogne slika 5, na kojoj je prikazano stablo rekurzivnog pozivanja procedure growtree

slika 4:

```
  3 A 0 0
  2 B 1 0
  1 A 1 1
  0 B 2 1
Konacni skor:  4 -  1
  0 A 1 2
Konacni skor:  1 -  4
  1 B 2 0
  0 A 2 1
Konacni skor:  2 -  3
  0 B 3 0
Konacni skor:  5 -  0
```



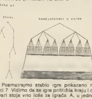
slika 5:



Minimiziranja i maksimiranja

Pošto smo u stanju da sastavimo program koji će generisati stablo neke igre, treba da formulišemo kriterijume na osnovu kojih će se računar odlučivati kojom od grana tog stabla da krene. Kada se igraču od krvi i mesa postavi problem tog tipa, on će razmotriti nekoliko svojih poteza, zatim nekoliko odgovora protivnika na neki od tih poteza i, u krajnjem slučaju, nekoliko svojih odgovora na te odgovore. Obzirom da su njegova mogućnosti procesiranja podataka vrlo ograničene i da mu je memorisanje velikog broja pozicija praktično nemoguće, čovek mora da ispolji selektivnost: izabraće vrlo mali broj perspektivnih poteza čije će stablo dalje ispitivati. Kompjuter, sa druge strane, ne poseduje intuiciju koja bi mu omogućila da odmah odbaci sve poteze koji će ga dovesti u inferioran položaj. Za početak ćemo, dakle, pisati programe koji generišu i ispituju sve moguće kombinacije poteza do određene dubine, rezultati takvoga rada su prikazani na slici 6

slika 6:



Posmatrajmo stablo igre prikazano na slici 7. Vidimo da se igra približila kraju i da stvari stoje vrlo loše za igrača A, u jednoj od grana ga, doduše, očekuje spektakularni dobitak od +256 poena, ali su ostale alternative gubitničke (jasno je, uzgred budi rečeno, da pozicija koja odgovara stablu sa slike 7 u igri „Poslednji pobeđuje?" ne postoji ali će sasvim lepo poslužiti onome što želimo da objasnimo). Pokušajmo da izaberemo jedan od tri poteza koje A može da odigra sledeći isključivo logiku bilsku kompjutersku

slika 7:



Sva tri poteza koja bi B odigrao u poziciji X vode do remija, što znači da bi poziciji X bilo umesno pridružiti vrednost 0

Sva tri poteza koja bi B odigrao u poziciji Y dovode do gubitka za A, pitanje je samo sa kolikom razlikom B dobija. Jasno je da će poziciji Y biti pridružen neki negativan broj ali ćemo se odlučiti koji je to broj tek kada razmotrimo poziciju Z

U poziciji Z igrač B ima na raspolaganju tri poteza od kojih jedan vodi u spektakularnu pobedu igrača A. Razmatrajući poziciju Z igrač A treba da razmotri kolika je verovatnoća te pobede. Ako A smatra da će B igrati potpuno slučajno (na primer bacajući kockicu), bilo bi logično usvojiti $(-9+(-4)+256)/3 = 81$ za vrednost pozicije Z što znači da bi A svakako izabrao potez Z a ne potez X. Međutim, rekli smo da razmatramo igre u kojima boginja Fortuna nema mnogo uticaja na tok stvari: B će razmotriti poziciju i izabrati potez koji mu bude izgledao najperspektivniji. Ne mora se, jasno, dogoditi da to bude i objektivno najbolji potez, ali je najsigurnija pretpostaviti da će B odigrati onaj potez koji bi A odigrao da je na njegovom mestu. To znači da ćemo poziciji Z dodeliti vrednost −9 tj. minimum vrednosti −9, −4 i +256. Na sličan način poziciji Y dodeljujemo minimum vrednosti −2, −1 i −7 tj. broj −7

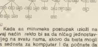
Tehnika minimaks

Pošto smo razmotrili pozicije X, Y i Z izabraćemo onu koja nam nudi *najveći mogući* skor pri optimalnoj igri protivnika. Obzirom da je maksimum brojeva 0, −7 i −9 upravo 0, odlučićemo se za potez X koji nas vodi do nerešene partije. Da bi sve bilo sasvim jasno, ponovićemo aksiome koje računar treba da sledi da bi izabrao najbolji mogući potez sledeći takozvani minimaks postupak

1. Vrednost pozicije u kojoj je A na potezu je *maksimum* vrednosti pozicija koje se dobijaju odigravanjem svakog od mogućih poteza

2. Vrednost pozicije u kojoj je B na potezu je *minimum* vrednosti pozicija koje se dobijaju odigravanjem svakog od mogućih poteza

Znajući ova dva aksioma, pogledajte ponovo stablo igre „Poslednji pobeđuje" sa slike 2 i pridružite svakoj poziciji njenu vrednost: dobićete sliku 8



Kada se minimaks postupak izloži na ovaj način, reklo bi se da ničeg jednostavnijeg na svetu nema, skoro da biste mogli da sednete za kompjuter i da počnete da pišete novi Sajrus šah. Pre nego što sastavimo program koji ilustruje minimaks postupak, dužni smo, međutim, da izložimo i njegove nedostatke koji ga čine neupotrebljivim kod iole složenijih igara

Činjenica da se svakoj poziciji primenom minimaks metode pridružuje konačan broj koji predstavlja njenu vrednost znači da je vrednost svake pozicije unapred odlučena: ukoliko je pozicija dobila vrednost +5, A će sigurno pobediti sa bar pet poena prednosti što znači da partiju nema mnogo smisla igrati. Obzirom da je šah jedna od igara za koju *je moguće* sastaviti minimaks stablo, javite Karpovu i Kasparovu da se bez potrebe muče igrajući mečeve! Nevolja je, međutim, što je za primenu minimaks postupka potrebno posedovati stablo *čitavog* toka igre da bi se odredila vrednost neke pozicije. Možete da zamislite da je stablo šahovske igre *prilično* (sli koliko?) veliko i da ga ne bi bilo lako nacrtati. Možda bi ipak vredelo probati? Pre nego što se odlučite za taj pokušaj treba da znate da šahovsko stablo ima oko 10E120 pozicija. Imate li predstavu o ovom džinu od broja? U jednom veku ima samo 10E16 *mikrosekundi*, a za svaki od tih mikrosekundi čak i najbrži kompjuter današnjice može da izvrši oko 1000 operacija kao što je sabiranje i oduzimanje. Kada bi se samo jednim sabiranjem mogla generisati nova šahovska pozicija, za ispitivanje 10E120 pozicija bi trebalo oko 10E100 *vekova*. A vasiona je stara mnogo manje *sekundi*! Vrlo je, dakle, neverovatno da ćemo u zamislivoj budućnosti imati priliku da procenimo početnu poziciju šahovske igre; kada to budemo mogli, očekujte šahovski problem na

koma je nacrtana početna pozicija i ispod koga piše bali vuča i dobija u najviše 696 poteza

Minimaks na delu

Iako je zahvaljujući poslednjim rečanicama minimaks postupak izgubio prilično od svoje atraktivnosti za sada je jedini koji poznajemo pa ćemo ga primeniti. Sastavićemo funkciju koja bira najbolji mogući potez u igri „Poslednji dobija?" koristeći sledeći algoritam

1 IF krajnja pozicija THEN nađi rezultat ELSE

2 FOR svaki mogući potez

3 Nađi vrednost pozicije koje je dobijena tim potezom

4 Pamti najbolju (max ili min) nađenu vrednost

5 NEXT potez

6 Najbolja nađena vrednost je vrednost funkcije

slika 9:

```
[illegible]
```

Na slici 9 je prikazana paskal funkcija koja vraća vrednost zadate pozicije. Primetimo da je funkcija veoma slična proceduri sa slike 3 koja samo generiše stablo mogućih pozicija. Startovanjem programa sa slike 9 treba da dobijamo vrednost pozicija 5A00 koju već znamo +1

Iako program sa slike 9 lepo radi, njegova struktura početnicima možda nije jasna jer se zavisno od vrednosti promenljive potez, traži minimum ili maksimum. Zato nam se čini razumnim pisanje dve odvojene funkcije koje će biti pozivane u slučajevima u kojima je A odnosno B na potezu i koje će biti daleko jednostavnije. Ove dve rekurzivne funkcije su prikazane na slici 10. *maxval* u stvari procenjuje vrednost neke pozicije kada je A na potezu, dok se *minval* poziva u cilju određivanja vrednosti pozicije u kojoj treba da igra B

slika 10

```
[illegible]
```

Sastavivši funkcije koje procenjuju vrednost pojedinih pozicija, dečje je igra napisati program koji će naučiti vaš kompjuter da igra „Poslednji dobija?", jedno od mogućih rešenja je prikazano na slici 11. Vredno je jedino ukazati na razliku između funkcija bestmove i maxval: maxval nalazi *vrednost neke pozicije*, dok bestmove u nekoj poziciji bira *najbolji mogući potez* pozivajući, naravno, funkciju maxval koja sa svoje strane poziva minval

slika 11:

```
[illegible]
```

Pre nego što, u „Računarima 10", pređemo na složenije igre, pomenimo mogućnost sastavljanja takozvanih tablica odluke. Umesto da se program muči i pronalazi najbolji mogući potez u svakoj poziciji, mogli bismo da ga snabdemo tabelom u kojoj bi, za svaki mogući broj šibica manji od nekog unapred predviđenog maksimuma, pronalazio podatak o broju palidrvaca koja treba da uzme da bi došao u dobijenu poziciju. Ovu tabelu bismo, jasno, mogli da sastavimo i ručno, ali je daleko lakše poslužiti se programom sa slike 12, koji poziva ranije sastavljenu funkciju bestmove. Daleko zanimljiviji rezultat bismo dobili ako u početku upišemo nule u tablicu, a zatim modifikujemo funkciju bestmove prema slici 13 — dobićemo program koji, kako igramo sa njim, uči i postaje nepobediv!

Dejan Ristanović

slika 12:

```
var table: array [1..10] of integer;
    Apoints,Bpoints:integer;

procedure setuptable;
var brojac:integer;
begin
   Apoints:=0; Bpoints:=0;
   for brojac:=1 to 9
      table [brojac]:=bestmove(brojac)
end;
```

Računari i matematika

ne diraj moje krugove

viša matematika uz računar — bolji računar uz matematiku

Našu šetnju kroz matematiku počinjemo druženjem sa konusnim presecima koji fascinira ju svojom jednostavnošću ali i činjenicom da nisu samo rezultati matematičkih apstrakcija već ih srećemo svuda oko sebe u svakodnevnom životu i koristimo ih za opise i dokaze mnogih realnih fizičkih fenomena. Za nas su posebno značajni stoga što njihovom rotacijom dobijamo iluziju trodimenzionalnog prostora pa ih možemo efektno iskoristiti za pisanje programa sa dobrom trodimenzionalnom grafikom.

Preseci i njihovo crtanje

Četiri na prvi pogled sasvim različita kriva: *krug, elipsa, parabola i hiperbola dobijaju se na isti način — presecima konusne površi nekom ravni* (sl. 1). Prvi put ih je detaljno opisao grčki matematičar Apolonije Pergajski (262—190. g.p.n.e.) iz grada Perga u Maloj Aziji u svom delu *Konusni preseci*, kojim je uticao na razvitak astronomije, mehanike, optike i dao osnove za stvaranje analitičke geometrije.

U osnovi cele priče je par pravih koje se seku i rotiraju oko svoje ose simetrije formirajući tako dvostruki konus. Te prave se zovu izvodnice (generatrise) konusne površi a ugao među njima — ugao konusa. *Ako konus presečemo sa ravni normalnom na osu simetrije, u preseku dobijamo krug. Ako ravan preseka seče sve izvodnice dobijamo elipsu.* U slučaju da je ova ravan *paralelna sa jednom od izvodnica dobijamo parabolu* i konačno ako je *presečna ravan paralelna dvema izvodnicama dobijamo dvogranu krivu koja se zove hiperbola* (sl. 2). Postoje i dva specijalna slučaja. Ako ravan preseka sadrži teme konusa i dve izvodnice, presek predstavljaju dve prave koje se seku što čini specijalan slučaj hiperbole. Konačno, ako ova ravan sadrži samo teme konusa, presek je tačka, odnosno krug degenerisan u tačku.

Preporučujemo da uz model načinjen od hartije vizuelno pojačate svoje predstave o konusnim presecima pre no što se upustimo u njihovo crtanje.

Da biste na računaru nacrtali grafike ovih krivih na najjednostavniji način, postavićemo ih u karakteristične položaje u odnosu na koordinatni početak i tako dobiti za njih najjednostavnije formule. Potrudićemo se da postignemo najbolje korišćenje TV ekrana, a ako kasnije želite da ove slike iskoristite u sopstvenim programima, promenom faktora uvećanja (M) dobićete odgovarajuću veličinu krivih. Morate voditi računa da pri translaciji likova ostanete u okviru ekrana, a posebno pazite i pri rotaciji parabole i hiperbole. Najbolje bi bilo da uvek proverite rezultat svojih izmena programa. Autor teksta je za ilustraciju ove teme preuzeo gotove programe za računare „spektrum" i „komodor", pri čemu programi za komodor zahtevaju sajmon's bejzik interpretator.

Krug

Svima nam je poznato da se *skup tačaka u ravni jednako udaljenih od stalne tačke 0 te ravni naziva krug*. Tačku 0 nazivamo *centar kruga*, a rastojanje ma koje njegove tačke od centra — *poluprečnik*.

Jednačinu kruga sa centrom u koordinatnom početku i poluprečnikom jednakim a možemo zapisati na sledeći način:

X=a*COS Θ
Y=a* SIN Θ

gde su (X,Y) koordinate bilo koje tačke sa kružnice, a Θ (theta) ugao između odgovarajućeg poluprečnika i x-ose. Program crta krug poluprečnika a sa centrom na sredini ekrana (sl. 3). Modifikujte program za crtanje kruga uvođenjem ciklusa u kome će se menjati vrednost parametra a i dobićete familiju koncentričnih krugova.

„spektrum"

```
10 CLS
15 LET a=70
25 LET x=a: LET y=0
30 PLOT 127+x,70+y
40 FOR t=0 TO 2*PI STEP .2
50 LET x=a*COS t: LET y=a*SIN t
60 DRAW x-PEEK 23677+127,
   y-PEEK 23678+70
70 NEXT t
```

„komodor"

```
10 HIRES 0,1:COLOUR 1,1
15 A=60
20 C=ATN(1)/45
30 XX=160+A:YY=100
40 FOR TH=0 TO 360 STEP 10
50 X=A*COS(TH*C):Y=A*SIN(TH*C)
60 LINE XX,YY,160+X,100+Y,1
65 XX=160+X:YY=100+Y
70 NEXT TH
80 GOTO 80
```

Elipsa

Skup tačaka u ravni čiji je zbir rastojanja od dve stalne tačke F_1 i F_2 te ravni konstantan zove se elipsa. Tačke F_1 i F_2 nazivamo *žiže* elipse. Ako rastojanje među žižama elipse, koje se zove *fokalno rastojanje*, označimo sa 2c a odgovarajući zbir rastojanja sa 2a, onda je očevidno $c/a=e<1$. Veličinu e nazivamo *ekscentricitet*. Prava koja sadrži žiže elipse seče elipsu u tačkama A_1 i A_2 koje su na međusobnom odstojanju 2a. Duž A_1A_2 zovemo *velika osa elipse*. Simetrala duži A_1A_2 seče elipsu u tačkama B_1 i B_2 koje su na međusobnom odstojanju 2b ($b^2=a^2-c^2$). Duž B_1B_2 zovemo *mala osa elipse*. U preseku velike i male ose nalazi se *centar 0* elipse. Ako nacrtamo elipsu sa centrom u koordinatnom početku i osama na koordinatnim osama, dobijamo jednačinu veoma sličnu jednačini kruga. Jednačina elipse se u ovom slučaju može zapisati na sledeći način:

X=a* COSΘ
Y=b*SIN Θ

Spljoštenost elipse možemo menjati promenom dužine poluosa a i b. Ako u programima sa slike 3 izvršimo sledeće izmene (slika 4)

„spektrum"

```
16 LET b=40
50 LET x=a*COS t: LET y=b*SIN t
```

„komodor"

```
16 B=30
50 X=A*COS(TH*C):Y=B*SIN(TH*C)
```

dobijamo na ekranu sliku elipse. Modifikujte program za crtanje elipse uvođenjem ciklusa u kome će se menjati veličina parametara a i b i dobićete uvid u zavisnost elipse od dužine poluosa. Iz ovog primera možete videti da krug predstavlja specijalan slučaj elipse ako je a=b. Dobro je imati na umu i sledeću činjenicu. Prave $x=\pm a/e$ koje su paralelne maloj osi na rastojanju a/e od nje zovu se *direktrise* elipse. Primetimo da odnos rastojanja ma koje tačke elipse od žiže prema rastojanju od odgovarajuće direktrise iznosi e, što nam daje mogućnost da elipsu definišemo kao skup tačaka za koje je odnos rastojanja od neke stalne tačke i rastojanja od neke stalne prave konstantna veličina (e) manja od 1.

Parabola

Skup tačaka u ravni jednako udaljenih od fiksirane tačke F te ravni i fiksirane prave d te ravni zove se parabola. Tačku F nazivamo *žiža* a pravu d *vodilja (direktrisa)* parabole. Presek prave koja prolazi kroz žižu a upravna je na vodilju nazivamo *teme parabole*. Rastojanje p temena parabole od žiže ili direktrise zove se *parametar parabole*. Ako postavimo parabolu tako da joj se teme nalazi u koordinatnom početku a žiža na pozitivnoj poluosi Ox, njenu jednačinu možemo zapisati na sledeći način:

$X=t^2$
$Y=2\times t$

Program dat na sl. 5 crta deo parabole koji se dobija promenom parametra t od −2 do 2. Proverite šta se dešava promenom faktora uvećanja označenim sa M.

„spektrum"

```
10 CLS
16 LET m=20
25 LET x=4*m: LET y=-4*m
30 PLOT 127+x,80+y
40 FOR t=-2 TO 2 STEP .05
50 LET x=m*t*t: LET y=2*m*t
```

Kada se zasitite video igara i dođete do one etape u korišćenju računara kada se postavlja pitanje „šta dalje da radim s njim — suviše je skup da čami u nekoj fioci, a nemam ni smisla ni volje da ga koristim ni za kakav „biznis", možda nije na odmet da se setite da su prvi računari konstruisani za vršenje složenih i dugotrajnih računanja. Nije li logično da svoj kućni računar počnete da koristite kao sredstvo koje će vam olakšati upoznavanje matematike? Mada za ponekog predstavlja velike poteškoće, ona je ipak kraljica nauke i onima koji se njome makar i s vremena na vreme bave pruža, pored mnoštva koristi, zbog čega se, uostalom, i uči u svakoj školi, ogromno intelektualno zadovoljstvo. Računari mogu pomoći i onima koje priroda nije obdarila talentom za matematiku da bez kompleksa savladaju barem srednjoškolsko gradivo. S druge strane bolje poznavanje matematike omogućiće vam da od svog računara dobijete mnogo, mnogo više no što vam se, možda do ovog trenutka činilo

```
60 DRAW x—PEEK 23677 + 127
   30+y—PEEK 23678+90
70 NEXT t
```

„komodor"

```
10 HIRES 0,1 COLOUR 1,1
15 M=23
20 C=ATN(1)/45
30 XX=160+M*4 YY=100—4*M
40 FOR T=—2 TO 2 STEP 05
50 X=M*T↑2 Y=2*M*T
60 LINE XX,YY,160+X,100+Y,1
65 XX=160+X YY=100+Y
70 NEXT T
80 GOTO 80
```

Hiperbola

Skup tačaka u ravni za koje je razlika odstojanja od dve fiksirane tačke F_1 i F_2 te ravni stalna zove se hiperbola. Ako kao kod elipse rastojanje među žižama označimo sa 2c a stalnu razliku *rastojanja* sa 2a onda vidimo da odgovarajuća veličina *e=c/a (ekscentricitet) hiperbole mora biti veća od 1.* Prave F_1F_2 zove se *fokalna ili presečna osa hiperbole* a simetrala duži F_1F_2 zove se *nepresečna osa.* Tačke A_1 i A_2 u kojima se grane hiperbole seku sa fokalnom osom nazivamo *temena hiperbole* a rastojanje među njima — *dužina fokalna (stvarna) osa.* Ako postavimo hiperbolu tako da se njena presečna osa poklapa sa x osom a nepresečna sa y osom onda njenu jednačinu možemo zapisati u sledećem obliku

X=a/COS Θ
Y=b* TAN Θ

Jedne grane hiperbole dobije se kada se Θ kreće od —90° do 90° a druge kada se Θ kreće od 90° do 270°. U ovim tačkama funkcija za X nije definisana jer je COS Θ=0. Mada bi se teoretski program i mogao rešiti samo jednim ciklusom u kome bi se Θ menjao od —90° do 270°, zbog probleme deobe sa nulom koji računar nije u stanju da reši koristimo dva ciklusa za crtanje grana hiperbole. I u ovom programu (slika 6) veličina M predstavlja faktor uvećanja pa predlažem da eksperimentišete sa njenom promenom.

Krug, elipsa parabola i hiperbola, dobijaju se na isti način — presecima konusne površi nekom ravni

„spektrum"

```
10 CLS
15 LET m=30
25 LET x=m/COS—1 LET y=m*TAN—1
30 PLOT 127+x,75+y
40 FOR t=—1 TO 1 STEP .1
50 LET x=m/COS t LET y=m*TAN t
60 DRAW 127+x—PEEK 23677,
   75+y—PEEK 23678
70 NEXT t
75 LET x=m/COS(PI—1)
   LET y=m*TAN(PI—1)
80 PLOT 127+x, 75+y
90 FOR t=PI—1 TO PI+1 STEP 1
100 LET x=m/COS t LET y=m*TAN t
110 DRAW 127+x—PEEK 23677,
   75+y—PEEK 23678
120 NEXT t
```

„komodor"

```
10 HIRES 0,1 COLOUR 1,1
15 M=60
20 C=ATN(1)/45
25 X=M/COS(—60*C) Y=M*TAN(—60*C)
30 XX=267 YY=0
40 FOR TH=—60 TO 60 STEP 5
50 X=M/COS(TH*C) Y=M*TAN(TH*C)
60 LINE XX,YY,160+X,100+Y,1
65 XX=160+X YY=100+Y
70 NEXT TH
75 X=M/COS(120*C) Y=M*TAN(120*C)
80 XX=50 YY=8
90 FOR TH=120 TO 240 STEP 5
100 X=M/COS(TH*C) Y=M*TAN(TH*C)
110 LINE XX,YY,160+X,100+Y,1
115 XX=180+X YY=100+Y
120 NEXT TH
130 GOTO 130
```

i za hiperbolu prave x=a/e i x=—a/e predstavljaju direktrise, pa i hiperbolu možemo označiti kao krivu za čije je tačke odnos rastojanja do fiksne tačke — žiže i rastojanje do fiksne prave — direktrise (vodilje) stalan i veći od 1 (e<1)

Rotacija krivih

U prethodnim programima smatrali smo da je x horizontalna a y vertikalna osa. Slika 7 ilustruje šta bi se desilo kada bismo rotirali elipsu za ugao AN. Tačka P bi se pomerila sa pozicije (X Y) na poziciju (XT YT) i te nove koordinate računali bismo po formulama

XT=X×COS AN — Y×SIN AN
YT=X×SIN AN+Y×COS AN

Navodimo potprogram (slika 8) pomoću koga bi se svaka od slike rotirala za ugao AN u odnosu na x osu



„spektrum"

```
1000 LET xt=x*COS (an*PI/180) —
   y*SIN (an*PI/180)
1010 LET yt=x*SIN (an*PI/180) +
   y*COS (an*PI/180)
1020 RETURN
```

„komodor"

```
1000 XT=X*COS(AN*C)—Y*SIN(AN*C)
1010 YT=Y*COS(AN*C)+X*SIN(AN*C)
1020 RETURN
```

Da biste sliku svake od navedenih krivih dobili u novom položaju treba da unesete neke izmene i dopune u odgovarajuća programa U liniji 17 definisana je vrednost ugla rotacija AN=60° a vi možete eksperimentisati i sa drugim uglovima Naročito je interesantno u okviru istog programa, predvideti da se kroz ciklus u kome bi se menjao ugao rotacije crtaju kriva u više položaja Tako dobijate uvid u neka rotaciona tela koja imaju mnogostruka primena Ne zaboravite da programu za svaku krivu pridružite i potprogram za rotaciju

Na slici 9 data su izmena za elipsu slici 10 za parabolu i slici 11 za hiperbolu

„spektrum"

```
17 LET an=60
20 GOSUB 1000
30 PLOT 127+xt,70+yt
55 GOSUB 1000
60 DRAW xt—PEEK 23677+127,
   yt—PEEK 23678+70
90 STOP
```

„komodor"

```
17 AN=60
25 X=A GOSUB 1000
30 XX=160+XT YY=100+YT
55 GOSUB 1000
60 LINE XX,YY,160+XT,100+YT,1
65 XX=160+XT YY=100+YT
```

„spektrum"

```
17 LET an=60
20 GOSUB 1000
30 PLOT 127+xt,90+yt
40 FOR t=—1 75 TO 1 75 STEP 05
55 GOSUB 1000
60 DRAW 127+xt—PEEK 23677,90+
   yt—PEEK 23678
90 STOP
```

„komodor"

```
17 AN=60
20 X=M*4 Y=—M*4 GOSUB 1000
30 XX=160+XT YY=100+YT
55 GOSUB 1000
60 LINE XX,YY,160+XT,100+YT,1
65 XX=160+XT YY=100+YT
```

„spektrum"

```
17 LET an=60
20 GOSUB 1000
30 PLOT 127+xt,75+yt
65 GOSUB 1000
60 DRAW 127+xt—PEEK 23677,
   75+yt—PEEK 23678
76 GOSUB 1000
60 PLOT 127+xt,75+yt
105 GOSUB 1000
110 DRAW 127+xt—PEEK 23677,
   75+yt—PEEK 23678
130 STOP
```

„komodor"

```
17 AN=60
20 GOSUB 1000
30 XX=160+XT YY=100+YT
55 GOSUB 1000
60 LINE XX,YY,160+XT,100+YT,1
65 XX=100+XT YY=100+YT
70 GOSUB 1000
60 XX=INT(160+XT) YY=INT(100+YT)
105 GOSUB 1000
110 LINE XX,YY,160+XT,100+YT,1
115 XX=160+XT YY=100+YT
```

Praktična primene

Kao što krug i telo koja se dobija njegovom rotacijom oko svog prečnika — lopta — imaju nebrojeno mnogo primena u svakodnevnom životu, tako i oblik elipse nije retka pojava u prirodi Planete se, recimo, kreću po eliptičnoj putanji u čijoj je jednoj žiži Sunce Jedna interesantna osobina elipse omogućava da se regulacija protoka tečnosti ili gasa u cevima vrši pomoću „leptira" eliptičnog oblika Naime pod određenim uglom elipse se projektuje u krug pa i ovi eliptični leptiri pod određenim pritiskom dolaze u položaj kojim u potpunosti zatvaraju cev Svojstvo da zraci koji prolaze kroz jednu žižu elipse po odbijanju prolaze kroz drugu žižu našlo je pored ostalih i svoju primenu u optici

Putanja po kojoj se kreću komete ili kosi hitac je paraboličnog oblika I parabola ima



krma

tzv optičko svojstvo — svi zraci koji je pogađaju paralelno osi odbijaju se kroz žižu a važi i obrnuto svi zraci koji prolaze kroz žižu odlaze sa parabola paralelno njenoj osi To svojstvo koje ima i telo koja nastaje rotacijom parabole oko svoje ose — paraboloid — koristi se u mnoge svrha Tako na primer postavljanjem sijalice u žižu paraboloida dobijamo paralelni snop svetlosti Na tom principu rade paraboloidni reflektori a ovo svojstvo koristi se i kod automobilskih farova Ako zrake sunca prikupljamo paraboloidnim ogledalima u žiži postižemo temperatura kao u sunčanim pećima Paraboloidna antene na ovom principu primaju i šalju signala

Napomenimo još da se svojstva hiperbole koriste u sistemu radarske navigacije brodova

Isto za sve

Pojam direktrisa za elipsu i hiperbolu nismo uveli slučajno On nam služi da uvedamo sve konusne preseke samo jednom definicijom *Svaki konusni presek predstavlja skup tačaka u ravni za koje je odnos rastojanja od neke stalne tačke — žiže i neke stalne prave — vodilje (direktrisa) konstantan* Taj stalni odnos rastojanja zove se ekscentricitet *Ako je ε<1, konusni*

Rotacija elipse

FIG.





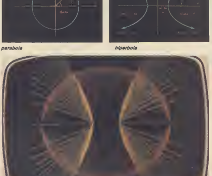
elipsa



parabola



hiperbola



Familije konusnih preseka sa raznim ekscentricitetima daje zanimljive grafičke efekte

 „spektrum"

```
10 BORDER 0 PAPER 0 INK 7 CLS
20 LET hyperbolae=90
30 LET ellipses=270
40 GOSUB hyperbolae
50 GOSUB ellipses
60 GOTO 60
80 LET ox=128 LET oy=87
90 FOR e=1 TO 2 STEP 0 05
100 LET a=22 LET b=a*(SQR (e∧2−1))
102 LET h=1
104 LET f=ox+(a/COS (−1 396))
106 LET g=oy+(b*TAN (−1 396))
108 IF g<0 THEN LET h=0
110 PLOT INVERSE 1, OVER 1,f,h
120 IF g>0 THEN PLOT INK 6,f,g
130 FOR t=−80 TO 60 STEP 20
135 LET r=t/(180/PI)
140 LET x=a/COS (r) LET y=b*TAN (r)
142 LET c=oy+y LET d=ox+x
150 IF h=0 THEN LET d=f+g*
   (f−d)/(c−g) PLOT d,h LET c=0
160 IF c>175 THEN LET d=d−
   ((d−PEEK23677)*(c−175)/
   (c−PEEK 23676)) LET c=175
170 DRAW INK 6,d−PEEK 23677,c−
   PEEK 23678 NEXT t
172 LET f=ox+(a/COS (1 75))
174 LET g=oy+b*TAN (1 75)
176 PLOT INVERSE 1, OVER 1,f,h
176 IF g<0 THEN LET h=0
180 IF g>0 THEN PLOT INK 6,f,g
190 FOR i=100 TO 260 STEP 20
195 LET r=t/(180/PI)
200 LET x=a/(COS (r)) LET y=b*TAN (r)
202 LET c=oy+y LET d=ox+x
204 IF h=0 THEN LET d=f+g*
   (f−d)/(c−g) PLOT d,h LET c=0
206 LET h=1
210 IF c>175 THEN LET d=0−
   ((d−PEEK 23677)*(c−175)/
   (c−PEEK 23676)) LET c=175
220 DRAW INK 6,d−PEEK 23677,
   c−PEEK 23678
230 NEXT t NEXT e
250 RETURN
270 FOR e=0 5 TO 0 88 STEP 0 04
280 LET a=100 LET b=a*
   (SQR (1−e∧2))
290 PLOT ox+a,oy
300 FOR t=0 TO 360 STEP 10
305 LET r=t/(180/PI)
310 LET x=a*COS (r)
320 LET y=b*SIN (r)
330 DRAW x−(PEEK 23677)+ox,
   y−(PEEK 23678)+oy
340 NEXT t NEXT e
360 RETURN
```

C= „komodor"

```
10 HIRES 1,6 MULTI 3,5,6
   COLOUR 0,0
20 C=ATN(1)/45
30 GOSUB 70
40 GOSUB 260
50 GOTO 50
70 FOR E=1 TO 1 50 STEP 04
100 A=13 8=A*SQR(E*E−1)
110 XX=80+INT(A/COS(−80*C))
   YY=100−INT(B*TAN(−80*C))
130 FOR TH=−80 TO 80 STEP 20
140 X=A/COS(TH*C)
150 Y=8*TAN(TH*C)
160 LINE XX,YY,80+X,100−Y,0+E
165 XX=80+X YY=100−Y
170 NEXT TH
180 XX=80+INT(A/COS(100*C))
   YY=100−INT(B*TAN(100*C))
190 FOR TH=100 TO 260 STEP 20
200 X=A/COS(TH*C)
210 Y=8*TAN(TH*C)
220 LINE XX,YY,80+X,100−Y,8+E
225 XX=80+X YY=100−Y
230 NEXT TH,E
250 RETURN
260 FOR E=45 TO 0 STEP −5
270 CIRCLE 80,100,35,E,3
280 NEXT E
290 RETURN
```

presek zove se elipsa, ako je $e=1$ — parabola, a ako je $e>1$ — hiperbola. I krug možemo da uvrstimo u ovu familiju definišući ga kao skup tačaka u ravni sa ekscentricitetom $e=0$

Sledeći program (slika 12) crta konusne preseke za razne ekscentricitete. Na početku menjamo e u oblasti $e>1$ i dobijamo familiju hiperbola. Zatim idemo u oblast $e<1$ i dobijamo familiju elipsi. Na „komodoru" elipse crtamo specijalnom naredbom CIRCLE. Preporučujemo da sastavite programe sa kombinovanjem bilo kojih konusnih preseka. Uverićete se da ćete dobiti i mnogo složenije i efektnije uzorke od onog na slici 13 koji proizvodi naš poslednji program.

Autor zahvaljuje prof. Branki Đerasimović na stručnoj pomoći u pripremi teksta.

Nevenka Spalević

Numerički metodi (I)

interpolacija - numerički krivuljar

Veoma je čest zadatak u kome je potrebno aproksimirati funkciju. To je situacija gde se za dato (izmereno)

$$x_0, x_1, x_2, \ldots, x_n \qquad (1)$$

znaju (mere) vrednosti neke funkcije f(x)

$$f(x_0)=y_0, f(x_1)=y_1, \ldots, f(x_n)=y_n \qquad (2)$$

a potrebno je korišćenjem tih podataka odrediti neku funkciju F(x) čije se vrednosti lako računaju i koja dobro aproksimira f(x). Kaže se da je F aproksimacija za f, i izračunavanje vrednosti f(x) vrši se korišćenjem F(x).

Aproksimacije i interpolacije

Vrednosti (1) i (2) mogu biti veličine određene u eksperimentu, testu i slično a vrednosti (1) često predstavljaju vreme. Ovakav zadatak se sreće u najvećem broju „diskretnih" eksperimenata, gde se veličine ne mere neprekidno. Zadatak aproksimacije se takođe javlja pri čitanju numeričkih tablica „između redova", ili zameni jedne funkcije drugom koja je bliska prvoj, a ima neke prednosti (recimo lakše se računa). Aproksimacija se može koristiti i pri predviđanju bliske „budućnosti" pojave koja se posmatra, kada se na osnovu poznatih vrednosti određuju vrednosti f(x) za $x > x_n$. Raznovrsnost primene metoda aproksimacija funkcija je jedan od razloga što se ova oblast ubraja u osnovne zadatke numeričke matematike.

Nije neophodno da F(x) uzima vrednost (2), odnosno prolazi kroz tačke date sa (1) i (2), već se pri aproksimaciji zahteva da F „malo odstupa" od f na intervalu. Međutim ako se postavi zahtev da F i f imaju baš iste vrednosti u datim tačkama, tj.

$$f(x_0)=F(x_0), \ldots, f(x_n)=F(x_n) \qquad (3)$$

a da pored toga F dobro aproksimira f u tačkama x različitim od tačaka (1), onda je to interpolaciona aproksimacija, ili kraće interpolacija.

Pri razmatranju interpolacije dovoljno je ograničiti se na razmatranje metoda za određivanje jedne vrednosti F(x) za proizvoljno x. Zato se zadatak interpolacije formuliše na sledeći način.

U tačkama (1) date su vrednosti (2). Treba izračunati nepoznatu vrednost f(x) u nekoj novoj datoj tački $x \neq x_i$, $i = 0, 1, \ldots, n$.

Ako je x u intervalu ograničenom najmanjom i najvećom tačkom iz (1) zadatak je interpolacija, a ako je x van tog intervala zadatak se naziva ekstrapolacija.

Ako je pored (1) i (2) umesto x data vrednost f(x) za koju onda treba odrediti pripadajuće x, zadatak se tada naziva inverzna interpolacija.

```
100 REM ****************************************
105 REM *                                      *
110 REM *     LINEARNA       INTERPOLACIJA     *
115 REM *                                      *
120 REM ****************************************
125 REM * Z-ULAZNA VREDNOST ARGUMENTA          *
130 REM *   INTERPOLACIJE                      *
135 REM * AI-NIZ VREDNOSTI NEZAVISNO           *
140 REM *   PROMENLJIVE (ULAZ)                 *
145 REM * VI-ULAZNI NIZ VREDNOSTI FUNKCIJE     *
150 REM * D-BROJ DATIH TACAKA (ULAZ)           *
155 REM * Y-INTERPOLIRANA VREDNOST FUNKCIJE    *
160 REM *   (IZLAZ)                            *
165 REM *                                      *
170 REM ****************************************
175 PRINT CHR$(147)
180 PRINT" ***********************************"
185 PRINT" * PROGRAM ZA INTERPOLACIJU FUNKCIJE *"
190 PRINT" *    JEDNE NEZAVISNO PROMENLJIVE    *"
195 PRINT" *                                   *"
200 PRINT" *       LINEARNA INTERPOLACIJA      *"
205 PRINT" ***********************************"
210 NI=5
215 GOSUB530:REM IZDAVANJE PRAZNIH REDOVA
220 INPUT" ( PRITISNITE RETURN DA NASTAVITE )";PR
225 PRINT CHR$(147)
230 DIM AI(100),VI(100)
235 REM ****************************************
240 REM *           ULAZNI PODACI
245 REM ****************************************
250 INPUT "UNESITE BROJ TACAKA";D
255 PRINT
260 PRINT"UNOSITE JEDAN (OPOD DRUGOG PAROVE"
265 PRINT"TACAKA  X(I),Y(I)"
270 PRINT
275 FOR L=1 TO D
280 PRINT"   X(";L;"),Y(";L;") =";
285 INPUT AI(L),VI(L)
290 NEXT L
295 PRINT
300 PRINT"AKO JE I-TI NEO POGRESAN, UNESITE I"
305 INPUT"AKO NEMA GRESKE, UNESITE 0";B
310 PRINT
315 IF B=0 THEN 345:REM TACNO UNETO, DALJE
320 IF B)0 THEN 295:REM VECE OD 0(NEMI)JE, PONOVO
325 PRINT"UNESITE X(";B;"),Y(";B;") =";
330 INPUT AI(B),VI(B):REM UNOSENJE ISPRAVKE
335 GOTO 295
340 PRINT
345 INPUT "UNESITE ARGUMENT INTERPOLACIJE";Z
350 PRINT
355 PRINT
360 PRINT
365 PRINT"*******  UNOS PODATAKA ZAVRSEN *******"
370 REM
375 REM ****************************************
380 REM *    ODREDJIVANJE POLOZAJA ARGUMENTA   *
385 REM ****************************************
390 IF Z>=AI(1) AND Z<=AI(D) THEN 415
395 PRINT
400 PRINT"  ???  ARGUMENT VAN OPSEGA  ???"
405 PRINT
410 GO TO 345
415 IF Z<>AI(1) THEN 430
420 Y=VI(1)
425 GOTO 470
430 FOR I=1 TO D
435 IF Z<=AI(I) THEN 460
440 NEXT I
445 REM ****************************************
450 REM *        LINEARNA INTERPOLACIJA        *
455 REM ****************************************
460 Y=(VI(I)-VI(I-1))*(Z-AI(I-1))
465 Y=Y/(AI(I)-AI(I-1))+VI(I-1)
470 GOSUB 565
475 REM ****************************************
480 REM *    KRAJ ILI NOVA INTERPOLACIJA       *
485 REM ****************************************
490 NI=5
495 GOSUB 530
500 INPUT"(1)-NOVA INTERPOLACIJA, (2)-KRAJ";B
505 IF B<>1 THEN END
510 GOTO 340
515 REM ****************************************
520 REM *     STAMPANJE NI PRAZNIH REDOVA      *
525 REM ****************************************
530 FOR N=1 TO NI
535 PRINT
540 NEXT N
545 RETURN
550 REM ****************************************
555 REM *  IZDAVANJE INTERPOLIRANE VREDNOSTI   *
560 REM ****************************************
565 PRINT
570 PRINT"INTERPOLIRANA VREDNOST FUNKCIJE"
575 PRINT"Y =";Y
580 PRINT
585 RETURN
```

Ne postoji jedan univerzalni metod koji rešava zadatak interpolacije u opštem slučaju. Drugim rečima, ne postoji metod koji bi za proizvoljnu funkciju f(x) rešavao

Mnogi konkretni problemi koji se rešavaju na računarima imaju, u svojoj osnovi, zadatke numeričke matematike. Numerički metodi poseduju veliku opštost i primenljivost, ali kako njihovo razumevanje i programiranje nije uvek sasvim jednostavno, to im upotreba odnosno popularnost među vlasnicima mikroračunara, nije na nivou koji zaslužuju. Ova serija napisa ima za cilj da jednostavno, i samo u najvažnijim elementima prikaže nekoliko metoda kojima se rešavaju neki osnovni zadaci numeričke matematike, kao i da ponudi programe koji su bazirani na tim metodama. Programi su pisani na bejziku komodor 64, ali se jednostavno mogu prenet na druge mikroračunare

zadatak interpolacije sa željenom tačnošću i sa jednostavnim oblikom F(x) koji omogućuje brzo izračunavanje. Otsustvo optimalnog metoda nije posledica nesistematskog pristupa ili nedovoljne razvijenosti numeričke analize već direktno izvire iz raznovrsnosti problema i njihove formulacije. Razvijeno je veoma mnogo metoda za interpolaciju i svaki od njih ima neke prednosti i mane u zavisnosti od osobina funkcije f(x) tipa funkcije F(x) tačnosti obima potrebnih izračunavanja, kao i pogodnosti pri prenošenju metoda na računare.

Interpolacija se uopšteno govoreći vrši tako što se konstruiše funkcija F(x) (linearna polinom, ili neka druga neprekidna funkcija) koja prolazi kroz sve date tačke određene sa (1) i (2). Time se naravno unosi greška koja se mora proceniti. U raznim metodama se koristi i različita količina informacija pri procenjivanju greške metoda. Negde u formulama za grešku figurišu i izvodi do visokog reda za funkciju f(x) što podrazumeva da su oni poznati a negde se za procenu greške koriste samo vrednosti koje su date u (1) i (2).

Linearna interpolacija

Ideja koja se intuitivno prva javlja u vezi sa zadatkom interpolacije je ideja linearne interpolacije, tj. aproksimiranje funkcije f(x) otsečcima prave na podintervalima

$$[x_i, x_{i-1}] \qquad i=0,1,\dots,n \qquad (4)$$

Ovim načinom se za F(x) uzima izlomljena poligonalna linija (neprekidno nastavljene duži) koja se u svojim tačkama prelamanja poklapa sa tačkama

$$(x_i, f(x_i)) \qquad i=0,1,\dots,n \qquad (5)$$

Jasno je i bez dublje analize da ova metoda ne može dati veliku tačnost aproksimacije, pogotovo kada se ima u vidu da se u linearnoj interpolaciji za određivanje vrednosti F(x) u nekoj tački podintervala (4) koriste samo vrednosti f(x) sa krajeva tog podintervala i ništa više (ostaje neiskorišćena informacija o vrednostima funkcije f u ostalim tačkama iz skupa (5)).

Međutim treba napomenuti da se ova metoda ne dobija samo kao najjednostavniji intuitivni pristup rešavanju zadatka interpolacije. Može se naime, dokazati da je to metoda do koje se dolazi ako se problem postavi na sledeći način.

Među svim neprekidnim funkcijama koje prolaze kroz tačke (5) naći onu koja je najglatkija (tj. onu koja minimizira funkcional zadat kao integral kvadrata prvog izvoda). Može se dokazati da je ta funkcija baš interpolaciona funkcija F(x) iz linearne interpolacije.

Ako je iz nekih razloga potrebna baš najglatkije aproksimaciona funkcija ili ako je potrebna prava najbrža i najjednostavnija interpolacija onda treba koristiti linearnu interpolaciju.

Linearna interpolacija vrši se prema formuli

$$F(x)=y_{i-1}+\frac{y_i-y_{i-1}}{x_i-x_{i-1}}(x-x_{i-1}) \qquad i=1,2,\dots,n$$

i ne zahteva ekvidistantnost tačaka (1) tj. one mogu biti na različitim međusobnim rastojanjima.

Navodimo sada bejzik program kojim se realizuje metod linearne interpolacije. Program nakon unošenja vrednosti (1) (2) i vrednosti interpolacionog argumenta x izdaje interpoliranu vrednost funkcije.

Kao i ostali programi koji će biti navedeni i ovaj je napisan tako da u dijalogu prima ulazne podatke omogućava njihovu eventualnu korekciju nakon unošenja a zatim izdaje interpoliranu vrednost. Na kraju se odgovorom na pitanje „Nova interpolacija ili kraj?" program može iskoristiti za računanje nove interpolirane vrednosti na osnovu istih podataka koji se ne moraju ponovo unositi. Na taj način se postiže lako dobijanje tačno onih interpoliranih vrednosti koje su potrebne.

Primer:
ulazni podaci
Broj tačaka = 5
(X(1),Y(1))=-1 1
(X(2) Y(2))=-0.5,0.25
(X(3) Y(3))=0,0
(X(4) Y(4))=0.5,0.25
(X(5),Y(5))=1,1
Argument interpolacije = 0.7
Izlaz
interpolirana vrednost = 0.55

Pretpostavimo da je u gornjem primeru zadatak bio da se linearnom interpolacijom na osnovu pet tačaka interpolira kvadratna funkcija

$$f(x)=x^2 \qquad (6)$$

Unete tačke (X(I) Y(I)) I=1 ... zaista odgovaraju kvadratnoj funkciji (6) a interpolirana vrednost 0.55 dobija se kao aproksimacija tačne 0.49. Ovo može da posluži kao ilustracija tačnosti koja bi se povećavala sa povećanjem broja tačaka (1) i (2).

Međutim, ako primer posmatramo kao rezultat interpolacije nepoznate funkcije čija su vrednosti dobijene merenjem onda bi tačnost zavisila od tipa nepoznate funkcije. Moglo bi se, čak, dogoditi da greška i nema, pod uslovom da je f(x) tipa deo-po-deo linearna funkcija što je teško očekivati u realnom eksperimentu.

Treba podvući da tačnost aproksimacije tj. interpolacije veoma zavisi od toga u kojoj meri upotrebljeni interpolacioni metod odnosno interpolaciona funkcija F(x) odgovara tipu funkcije f(x) koja se aproksimira.

Polinomijalna interpolacija

Kada se za interpolacionu funkciju F(x) uzme polinom onda se takva interpolacija naziva polinomijalna interpolacija. Ovo je veoma rasprostranjen vid interpolacije i u mnogim slučajevima daje dobre rezultate. Metod se bazira na sledećem.

Polazi se od polinoma tj. funkcije oblika

$$F(x)=a_0+a_1x+a_2x^2+\dots+a_nx^n \qquad (7)$$

koja se uzima za interpolacionu i postavlja se zahtev (3) da ova funkcija mora prolaziti kroz sve date tačke. Time se dobija sledeći sistem jednačina

$$\begin{aligned}
a_0+a_1x_0+a_2x_0^2+\dots+a_nx_0^n&=f(x_0)\\
a_0+a_1x_1+a_2x_1^2+\dots+a_nx_1^n&=f(x_1)\\
a_0+a_1x_2+a_2x_2^2+\dots+a_nx_2^n&=f(x_2)\\
&\vdots\\
a_0+a_1x_n+a_2x_n^2+\dots+a_nx_n^n&=f(x_n)
\end{aligned} \qquad (8)$$

koji predstavlja sistem od n+1 linearne jednačine sa n+1 nepoznatom

$$a_0, a_1, \dots, a_n \qquad (9)$$

Može se dokazati da pod pretpostavkom da su sve zadate tačke (1) međusobno različite sistem (8) ima rešenje koje je jedinstveno. Time se zapravo dokazuje da traženi interpolacioni polinom (7) postoji i da je jedinstven a njegovi nepoznati koeficijenti se mogu odrediti rešavanjem sistema (8).

Pri izračunavanju tražene interpolirane vrednosti moglo bi se dakle, najpre korišćenjem datih vrednosti (1) i (2) formirati sistem (8) a zatim taj sistem rešiti i time dobiti koeficijente (9). Njih bi zatim, trebalo zameniti u izraz (7) za interpolacioni polinom. Time bi se formirao polinom koji za zadati interpolacioni argument x daje interpoliranu vrednost F(x).

Međutim polinom (7) se može transformisati u nekoliko različitih oblika koji omogućavaju direktno uzimanje u obzir uslova (3) tako da se izbegava rešavanje sistema (8). Najpoznatija su dva tipa tako zapisanog polinoma (7). To su Lagrangeov (Lagranžov) i Newtonov (Njutnov) interpolacioni polinom.

Ovim interpolacionim polinomima, greškama koje oni unose kao i programima koji vrše polinomijalnu interpolaciju biće posvećen naredni nastavak.

Mr Veljko Spasić

Računari u poslovnoj primeni — kadrovi na računaru — Obrada podataka u radnoj organizaciji



U prošlom broju prikazan je jedan od mogućih načina obrade ličnih dohodaka u radnoj organizaciji aplikacije koja danas postoji na svakom velikom računaru u našoj zemlji. Sada ćemo dati opis postupka obrade podataka koji postoje u svakoj radnoj organizaciji. Njihove obrade se vrši samo u malom broju računskih centara zbog toga što se one obično smatra sporednim poslom što u većini radnih organizacija ima tretman „nametnutog" rada i konačno, što se zbog veće ponude nego potražnje na sistematsku obradu podataka gleda kao na čisti luksuz. Pa ipak, dobro vođena kadrovska politika jedan je od preduslova za uspešno poslovanje. Tema ovog članka će biti kadrovska evidencija.

Projektni zahtev

Formirati matičnu datoteku za potrebe kadrovske evidencije, vodeći računa o zakonski propisanim minimumom podataka (Sl glasnik SRS br 1/80), kao i specifičnim potrebama same radne organizacije. Ovako formirana datoteka treba da omogući dobijanje najrazličitijih pregleda potrebnih kadrovskoj službi kao što su

— pregled radnika po kvalifikacionoj strukturi zaposlenih

— pregled kadrova koji su pred penzionisanjem (na primer, 3 ili 5 godina) prema kvalifikacijama

— pregled kadrova na stručnom usavršavanju

— izdvajanje radnika koji ispunjavaju različite zahteve (na primer znanje određenog stranog jezika, određena kvalifikacija, određeni staž i slično), ili pak izdavanje rešenja radnicima, na primer za

— godišnji odmor

— prestanak radnog odnosa

— primanje u radni odnos

— izdavanje novih rešenja o izmeni broja bodova ili novom radnom mestu

— pravo na minuli rad i slično

*Obradu izvršiti na personalnom računaru sa konfiguracijom koja je opisana u prethodnom članku**

Sadržaj datoteka

Sadržaj datoteke je jednim delom već određen zakonski propisanim minimumom i obuhvaćen je odgovarajućim kadrovskim upitnikom (KAD 1/1, KAD 1/2 i KAD 1/3). Ovde ćemo se zadržati na komentaru ovih obrazaca sa stanovišta potrebe radne organizacije.

Obrazac KAD 1/1 sadrži opšte podatke o radniku, kao i podatke o radnom odnosu. Prvo 21 mesto nije potrebno obuhvatati jer one ne sadrže nikakve relevantne podatke od interesa za radnu organizaciju. Obuhvatanjem matičnog broja građana otpada i potreba da se unosi podatak o datumu rođenja i polu radnika jer se oni već nalaze u matičnom broju. Iz dela podataka koji se odnose na radno angažovanje takođe prvih 21 znakova nije potrebno obuhvatati, dok svi ostali podaci mogu da se prenesu u našu matičnu datoteku.

Obrazac KAD 1/2 sadrži podatke o ranijem radnom odnosu kao i podatke o obrazovanju i stručnom usavršavanju. Podatke o ranijem radnom odnosu nije potrebno obuhvatati dok se druga grupa podataka može kompletno uvrstiti u našu datoteku.

Obrazac KAD 1/3 sadrži podatke o društveno-političkoj aktivnosti oblastima interesovanja i podacima od interesa za narodnu odbranu. Svi ovi podaci mogu biti uneti u matičnu datoteku.

Pored ovih podataka poželjno je da matična datoteka sadrži i podatke koji nisu obuhvaćeni ovim obrascima a predstavljaju određeni interes za radnu organizaciju. Ovde se u prvom redu misli na

— matični broj radnika u radnoj organizaciji, koji će ujedno služiti i kao ključ za pronalaženje podataka za određenog radnika

— šifra organizacionog dela u kome radnik radi (OOUR, radna jedinica i slično)

— datum kada radnik stiče novu, punu godinu radnog staža

— broj poslednjeg rešenja o raspoređivanju radnika

— broj bodova za lični dohodak

Posebnu grupu podataka predstavljaju podaci vezani za praćenje društvenog standarda zaposlenih. Ove podatke može dati samo odgovarajuće službe iz same radne organizacije ukoliko one postoji. U svakom slučaju, ako se ovi podaci unesu u matičnu datoteku, treba obezbediti i praćenje promena kako bi se blagovremenim ažuriranjem podataka dobile i korektno obrađene informacije. Ova opaska je data iz iskustva vezanog za preambiciozne planove odgovarajuće službe za praćenje društvenog standarda. Inicijalno unošenje najšireg skupa podataka moguće je izvršiti anketom ali je često sprovođenje takve ankete zbog održavanja podataka ažurnim ne samo skupo već i psihološki nepoželjno (teško je svakog meseca odgovarati na pitanje „Da li imate televizor u boji?" ili „Da li imate vikendicu?"). Stoga u ovom delu treba biti jako obazriv sa podacima koji se žele pratiti.

Postojanje ovakve datoteke od izuzetne je važnosti za organizovanje društvene samozaštite i teritorijalne odbrane. Za ove potrebe nužno je svakako uneti

— adresu stana radnika

— broj telefona (ukoliko je PTT mreža dovoljno razgranata da veći broj radnika ima telefon)

— raspored

— čin

* ... opisu konfiguracije u tekstu. Plavi koverti iz ra... greškom je u zagradi stavljeno poželjno što manje, a treba poželjno što više

Iako se u poslednje vreme za kućne računare nalazi i u kući poneki ozbiljniji posao, većina i dalje ostaje da čami neiskorišćena, a novopečeni programeri nisu ni svesni kakvo moćno oruđe imaju u rukama. Kućni računari, međutim, idealni su za malu privredu, a mogu, na određenim poslovima, da nađu svoje mesto čak i u najvećim radnim organizacijama, bez obzira da li u njima već postoji računarski sistem. U ovoj seriji tekstova analiziramo neke značajne mogućnosti primene ličnih računara za potrebe organizacija udruženog rada.

Olančavanje podataka

Na ovaj način formirana datoteka ima oko 600 bajtova dužine što je znatno više od 128 bajtova koliko se preporučuje pri radu sa ličnim računarima. Međutim, možemo se ovde poslužiti jednim malim trikom za olančavanje slogova koji se odnose na jednog radnika. Naime već smo rekli da je najpogodnije da pristup podacima za jednog radnika bude preko njegovog matičnog broja koji mu je dodeljen prilikom dolaska u radnu organizaciju. Ako se iza ovoga broja doda još jedna cifra, to svakom radniku omogućujemo da se njegovi podaci rasprostru na 10 slogova koji će slediti jedan iz drugog. Prvi slog sa svojih 128 bajtova može, na primer, da sadrži opšte i adresne podatke o radniku, drugi o obrazovanju i stručnom usavršavanju, a peti, recimo, o društveno-političkoj aktivnosti.

Pozivajući se, na primer, na matični broj radnika 1255 i dodajući, recimo, nulu (što radi sam računar), pristupamo prvom slogu, to jest ključu 12550. Nakon završetka obrade podataka iz ovog dela prelazimo na ključ 12551, to jest na drugi slog sa podacima za istog radnika, i tako redom, do poslednjeg sloga koji za osnovu ključa ima broj 1255. Posle toga program se vraća natrag, očekujući unos podataka o novom broju radnika. Prilikom sekvencijalne obrade ove datoteke moguće je odmah nakon učitavanja izvršiti selekciju da li su nam potrebni podaci iz tog dela ili nisu. Ukoliko nisu, idemo na čitanje sledećeg sloga, a ukoliko jesu, na odgovarajuću obradu. Ovako olančani podaci u izvesnoj meri povećavaju prostor potreban za smeštaj slogova na disketu, jer se unutar svakog sloga pojavljuje matični broj radnika i dodatni indikator rednog broja sloga. Međutim, ovo povećanje dužine ne utiče bitno na broj slogova koji se mogu smestiti na jednu disketu.

Ako pretpostavimo da sve podatke koje želimo da pratimo možemo smestiti na 5 slogova dužine 128 bajtova, to znači da bismo na disketu od 300KB mogli uneti podatke za oko 400 radnika. Ukoliko imamo potrebu za većim brojem radnika, datoteku možemo podeliti na više disketa, tako da nismo ni na koji način ograničeni veličinom same datoteke. Pri sekvencijalnoj obradi, nakon završetka jedne diskete bismo postavili sledeću i tako do n-te diskete koliko ih zauzima naša datoteka. Pri direktnom pristupu podacima moguće je nakon unošenja matičnog broja radnika proveriti opseg u kome se željeni broj nalazi i dati poruku da se unese odgovarajuća disketa. Praktično, ograničeni smo jedino u slučaju da se zahtevaju neki pregledi koji su sortirani (na primer, po azbučnom redu imena i prezimena radnika).

U postupku izdvajanja slogova za izveštaj jedna disketa bi sadržala matičnu datoteku radnika, a druga izlaznu datoteku za izveštaj. Kako obično na ličnim računarima raspolažemo štampačem od 80 kolona, to je, s obzirom na razmake između pojedinih podataka u izveštaju, slog koji se štampa recimo 64 bajta, te je u 1KB moguće smestiti 16 takvih slogova. Pri sortiranju podataka potrebne su nam radne datoteke, te teorijski maksimum koji možemo obezbediti na disketi od 300KB iznosi dve radne datoteke od 150KB. Na taj način dolazimo do teorijskog maksimuma od 16 × 150 = 2400 slogova, odnosno radnika. Ovakva datoteka bi se prostirala na 8 disketa, međutim, ovako velike radne organizacije verovatno već imaju svoje velike računarske sisteme.

Napomenimo na kraju, da se ova datoteka obično koristi kao matična datoteka radnika za obračun ličnih dohodaka, s obzirom da sadrži neke podatke koji su i tamo prisutni (prezime i ime radnika, broj bodova po rešenju, kvalifikacije, članstvo u SK, opština i mesto stanovanja i slično). Ukoliko se želi koristiti ista datoteka i za obradu ličnih dohodaka, potrebno je pored već navedenih podataka u njen sadržaj uvrstiti i podatke koji se traže pri obradi ličnih dohodaka, a koji su navedeni u našem prethodnom članku.

Programski deo

Koji će programi biti potrebni za praćenje kadrovske evidencije veoma je teško reći unapred. Samo oni koji neposredno rade u ovim službama znaju koliko najrazličitijih zahteva dobijaju kako od pojedinih sektora iz sopstvene radne organizacije, tako i od raznih službi iz opštine ili republike. Sistematizovanje ovih izveštaja nas ne bi dovelo ni do kakvog rezultata, ali baš to čini da se posebna pažnja mora posvetiti samom sadržaju matične datoteke, kako bi mogli da zadovolji svi zahtevi koji se pred nju postavljaju.

Pa ipak, postoje neki programi koji čine izuzetak od ovoga. To su:

— program za inicijalno formiranje matične datoteke,

— program za ažuriranje podataka u matičnoj datoteci,

— programi za izdavanje odgovarajućih rešenja radnicima, pri čemu prva dva programa mogu biti spojena u jedinstven program.

Program za inicijalno formiranje matične datoteke treba da omogući unos podataka za trenutno zatečene radnike unutar radne organizacije, uz odgovarajuću logičku kontrolu podataka, koja je u ovom slučaju veoma skromna. Naime jedini podaci koji se mogu kontrolisati sa stanovišta logike su matični broj građana (po modulu) i datumi koji se pojavljuju unutar same datoteke (samo sa stanovišta realnosti datuma, a ne i njegove tačnosti). Stoga je preporučljivo da se u fazi formiranja ove datoteke vrši poređenje unetih podataka sa podacima iz izvornih dokumenata.

Program za ažuriranje podataka treba da omogući:

— unos novih podataka,

— ažuriranje postojećih podataka,

— brisanje slogova za radnika koji je napustio radnu organizaciju.

Deo za unos novih podataka (slogova za novoprimljenog radnika) u suštini je identičan onom koji je bio opisan za program pri inicijalnom formiranju datoteke. Ažuriranje podataka je poželjno rešiti preko odgovarajućeg „menija" koji se prikazuje na ekranu uz istovremeni prikaz, na primer u poslednjem redu ekrana, sadržaja koji se menja. Brisanje slogova vrši se na standardan način, s tim što je i ovde, radi kontrole, poželjno na ekranu pre brisanja prikazati bar ime i prezime radnika čiji se podaci žele uklониti.

U okviru ažuriranja posebnu grupu predstavljaju programi koji sami, nakon izdavanja odgovarajućeg izveštaja, ažuriraju podatke u matičnoj datoteci. Tako, na primer, program koji radniku izdaje obaveštenje o izmeni broja bodova po osnovu minulog rada vrši i ažuriranje staža radnika, a program koji izdaje radniku rešenje o novom radnom mestu ažurira broj bodova, šifru radnog mesta, broj poslednjeg izdatog rešenja i datum novog rešenja.

Programi za izdavanje odgovarajućih rešenja radnicima obuhvataju sva rešenja koja kadrovska služba izdaje radnicima u najrazličitijim slučajevima (godišnji odmor, izmene sistematizacije, promene startne osnove — broja bodova za lični dohodak, promene radnog mesta, i slično). U ovom slučaju moguće je štampu vršiti ili na unapred pripremljenim obrascima ili, pak, pustiti mašti na volju da „igrajući se" matričnim štampačem programer formira sam dokument. Kako se većina rešenja izdaje u više primeraka, potrebno je pre izrade programa proveriti mogućnost štampe na višestrukom papiru. Ukoliko se pokaže da kopije nisu odgovarajućeg kvaliteta, potrebno je programski obezbediti višestruko ponavljanje štampe istog dokumenta bez ponavljanja čitave obrade.

Komentar

Ovde je dat jedan način obrade podataka za kadrovsku evidenciju zasnovan na klasičnom načinu obrade preko matične datoteke. Postojanje baze podataka na nekim ličnim računarima svakako bi omogućilo sasvim nov pristup rešavanju ovog problema, ali za analizu ovakvog načina rada bilo bi potrebno usvojiti ne samo konfiguraciju sistema na kome se radi nego i određeni računar, te i sam programski paket koji podržava rad sa bazom podataka, što bi sa svoje strane veoma suzilo mogućnost primene.

Mihailo Karapandžić, dipl. inž.



Novi imidž za „nove" „Računare"

Iskreno da vam kažem bio sam oduševljen kad sam saznao da su Računari postali mesečnik Jeste da će se to osetiti na mom džeparcu ali ono što prija još više prija kad je češće Naravno imao bih i nekoliko predloga za ove nove Računare Manje je i do sad se svim odgovaralo kako vaš časopis izgleda ali smatram da bi se mesečnim izlaženjem mogli da promenite neke stvari Tako biste postali još bolji i verovatno popularniji Prvo bih vam predložio da malo razbijete ona velike i opsežne članke koji uglavnom zauzimaju središnu vašeg lista Oni jesu izvanredno kvalitetni ali bi bilo mnogo interesantnije i nekako primerenije mesečniku da ima više manjih rubrika Mogli biste takođe da insistirate na najsvežijim informacijama po potovu kad imate sjajnog dopisnika iz Londona a sad i iz Amerike Predložio bih vam takođe da još više pažnje posvetite grafičkoj opremi lista Ona jeste dobra ali vi imate jedan izuzetno dinamičan zanimljiv i kvalitetan časopis i njegova grafička oprema bi morala da bude izvanredna a ne samo dobra Nemojte da se uvredite što predlažem ovoliko izmena to je u najboljoj nameri

Vaš čitalac

Miodrag Marković
Niš

Hvala ti na korisnim i sasvim primenljivim savetima Oni se izvrsno uklapaju u razmišljanja čitave redakcije koja je trenutno u poslu prilagođavanje „Računara" novom, mesečnom imidžu Zato se napormo radili na temama i mislima da će ti, kako ti kažeš, „novi" „Računari" sadržati u sebi mnogo novog što odgovara tvojim predlozima

Umetci, umetci, i samo umetci

Šta sad kad ste objavili rutine za spektrum i komodor ? Da li to znači da neće biti više umetaka? Nemojte da vam se to desi Lepo ste počeli sa dodacima i sad nemojte da prestanete Znam da nemate spremljene rutine za amstrad ili čak neki atari ali to ne znači da ne biste mogli da objavite nastavak rutina za spektrum ili možda da počnete sa školom nekog jezika ili nešto slično Uostalom vi razmislite šta ćete Jeste li razmišljali da raspišete neki manji konkurs za scenario za video igre ili neki drugi tip programa? Znam da to deluje jako neprogramerski ali dar trojica mojih drugara-hakera bi mogli da naprave prava čuda od igara kad bi im samo neko rekao kakav treba da bude sadržaj Ako je to tako sa trojicom mojih drugara sigurno ih ima još takvih Zato razmislite malo o tome

Draškić Nedeljko
Cvijićeva 5
Osijak

Umetke planiramo za svaki drugi broj. Rutina za „amstrad" su, inače, potpuno spremljene Samo što nisu ušle u štampu Što se scenarističkog konkursa tiče, razmislićemo

„Galaksija" bolja od „atarija"?

Ima nešto jako čudno u Računarima 8 Dosta pažljivo sam pročitao prikaze računara i primetio sam da je odnos između dve sasvim različite mašine kao što su atari 520 ST i galaksija+ malo poremećen! Moguće je da je to samo moj utisak ali čini se kao da se Dejanu Ristanoviću više sviđa galaksija+ od atarija 520 ST Ja se verovatno ne razumem u računare toliko koliko Dejan ali čak je i meni sasvim jasno da atari čak i sa svim njegovim manama spada u klasu koja je svetlosnim godinama daleko od galaksija makar ona bila i + Ako je u pitanju lokal-patriotizam onda je on došao na malo pogrešnom mestu Naravno prikaz atarija i nije nešto detaljno dat pa je moguće da postoje neki podaci koje mi čitaoci ne znamo ali ako je tako onda je Dejan trebalo i njih da stavi u tekst Naravno to ništa ne menja u činjenici da je Dejan Ristanović za mene i dalje jedan od najboljih domaćih autora na području računara

Svaki od dva pomenuta kompjutera je prikazan i analiziran u odnosu na klasu računara kojoj pripada i poređenje između njih je potpuno apsurdno. Pošto su tvorci računara „atari 520 ST" ciljali dosta visoko nije nimalo čudno što je „džekintoš" pretrpeo i neke kritike. Dejanu se, razume se posle PC-a, „amige" i još nekih računara više sviđa „atari", ali ga to nimalo ne sprečava da navija za — „galaksiju"

Okrugli sto umesto „stručnjaka"

Javljam se ovim pismom zbog ličnog osećanja dužnosti a i ogorčenja Čitam Galaksiju već dugo vremena i zaista je pratim redovno U srednjoj školi sam bila programer i ono što je škola propustila u obrazovanju vaš i moj časopis je nadoknadio Trenutno studiram Višu školu za primenjenu informatiku i statistiku

Vaš časopis je na žalost postao poligon za preganjanja velikih i malih obrazovanih i onih koji to nisu Zašto? Verovatno ste to dozvolili ne upuštajući se u dalje tokove događaja Smatram se veoma pogođenom kada mi neki nazovi programer, informatičar ili stručnjak iz moje fele soli pamet Dakle da ne dužim mnogo predlažem da se umesto polemika i preganjanja preko novina lepo napravi jedan okrugli sto otvoren za sve Tu podrazumevam i đake studente profesore ljude koji već rade na takvim poslovima i one koje ove stvari interesuju Takav okrugli sto sigurno ne bi razrešio sve dileme ali bi dar bio neka vodilja a je za sada u računarstvu naše zemlje nisam ništa slično videla

Jedino složno mi možemo uvesti malo reda i mira u ovaj kovitlac Prosto mi je muka kad vidim ljude naduvene u svoje znanje kako drže katedre i stolice u radnim organizacijama

Nadam se da ćete i ovog puta biti prvi u organizovanju

Pozdravlja vas

Mirjana Nikolić
Beograd

Mirjana, tvoj predlog je izuzetno zanimljiv. Vrlo je verovatno da će redakcija, čim detaljno razradi kako bi to trebalo da izgleda organizovati nešto slično. Hvala i piši nam ponovo, pogotovu ako budeš imala još neki sličan predlog.

Imam jednu želju

Znate šta ja mogu da razumem da vi ne možete da dobijete Amigu ili Fat Mac pa da ih onda prvi prikažete spolja i iznutra To je sasvim razumljivo u pitanju su pare kriza para Balkan kao provincije para i tako te stvari Ali ono što biste vi mogli da uradite je da ukažete dar na one stvari koje se kod nas pojavljuju Kako to da nema prikaza knjiga u Računarima ? Pa bar biste to mogli da objavljujete — knjiga je sve više i sve su zanimljivija (i skuplje) Pošto mi čitaoci ne možemo daš da eksperimentišemo sa knjigama i da dajemo pare na gluposti bilo bi dobro da nas dar posavetujete šta valja a šta ne isto to biste mogli da učinite i sa softverom a ne samo sa uvoznim igrama S obzirom da ste sada mesečnik to valjda znači da imate više vremena i prostora da se malo priklonite željama čitalaca Puno pozdrava

Radmila Bunjevac
Cvijićeva 16/5
Zemun

Pa mi smo i do sada ispunjavali želje naših čitalaca Te-ne-nenem, tvoja želja je upravo ispunjena Od sad idu prikazi

Ni skupoća ni fuš

Pišem vam ovo pismo kao odgovor (moj a možda i mnogih kao što sam ja) na intervju sa dr Dušanom Slavičem tve bih želio da ovo moje javljanje ispadne kao neko prepucavanje Zašto se drug Slavić, kada je uvideo mnogo grešaka i propusta u nekim programima jednostavno nije javio redakcijama koje su te programe izdale Uvjeren sam da bi oni rado objavili ispravke Drugo kako da neki klinac koji je jedva nagovorio roditelje da mu kupe računar sazna nešto više o programiranju u najnovijim verzijama paskala i fortrana kada ga u školi uče kako su kompjuteri čitali bušene kartice Iskreno smatram da nije sve u životu izračunavanje logaritama množenja itd i da se svaki normalan čovek ako je ikada video da mu neko ispred nosa igra video igru i sam uhvati za palice za igru Smatram da je i dr Slavić ponekad tako učinio i da ne treba da se boji da ćemo pomisliti da je zbog toga podetinjio

Sada bih napisao nešto o Računarima Odlično ste počeli galaksija razni hardveri za spektrum i komodor 64 softver ali što dalje sve manje o tome a sve više apstraktnih tekstova i prikaza skupih i neadekvatnih modela (za naše tržište) Takođe mislim da je glupo davati opise naših domaćih računara jer oni nisu pravljeni za prosečnog Jugovića Ko je toliko lud da kupi neku fuš-mašinu (domeću) koja košta po 20 i više starih miliona (Svaka čast galaksiji za samogradnju) Takođe nema novosti o igrama Dobra vam je rubrika Hakeri u nevolji Sve ovo vam je napisao student prve godine Elektrotehnike uz pozdrav redakciji

Mirković Boris
Maršala Tita 42
56236 Ilok

Dragi Borisa, uglavnom se slažemo sa tobom Ono gde smo malo dovedeni u nedoumicu je tvoj savet o kojim računarima ne pisati Ako ne pišemo o stranim skupim računarima (za Jugoviće su svi strani računari skupi) i ako ne pišemo o domaćim „fuš-mašinama", kako kažeš, o čemu ćemo pisati? To su otprilike svi računari koji postoje Što se tiče dr Dušana Slavića, on od svih Računara" prelazi sa reči na dela

Dajte još legendi

Javljam vam se povodom jednog teksta koji je išao u Galaksiji a ne u Računarima U oktobarskom broju Galaksije objavili ste tekst o Džonu fon Nojmanu jednom od legendarnih pionira računarstva Moram vam priznati da me je tekst izuzetno zainteresovao Zašto ne bi bilo više takvih tekstova? Mogli biste da tu rubriku (Pioniri Računstva) predacite u Računare i da svakog meseca objavljujete kratku biografiju jednog od legendarnih računardžija (naravno, i računardžijki) Ja lično bih volse da pročitam nešto više o Adi Babidžovoj asistentici Toliko sam toga interesantnog čuo o njoj — uglavnom od drugova koji su pročitali neke knjige na engleskom

Čudno da se niko od nas nije setio da izda knjigu sa skraćenom istorijom računarstva? Šta kažete na tu ideju?

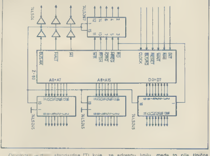
Milorad Purić
Zrenjanin

Računari u domaćoj radinosti

doping za z80

Ako se lako uzbudite kad osetite [illegible] toplotu lemilice [illegible] miris tinola a kad vidite shemu nekog novog kompjutera kroz telo vam pr[illegible] d[illegible] [illegible] vam se znoje a ruke da podrhtavaju, stvar je jasna Vašim venama teče krv hardv[illegible] [illegible] spremni ste da date sve od sebe da uništite tek kupljani kompjuter Ovaj — projakat ima zadatak [illegible] ušilački posao

[illegible]	struje kojom opterećuje izlaz procesora
[illegible] (74, 7442 itd)	1 6 mA
[illegible] (74 LS [illegible])	0 18 mA
[illegible]	10 µ A
[illegible]	10 µ A



Već i mala deca znaju da se [illegible] kompjutera dele na dva podvrste [illegible] še i gorepomenute uvrnute lipove [illegible] le hardverom mašina za računanje [illegible] prvi imaju muka sa izborom k[illegible] drugi tej problem nameju Svaki prav[illegible] dveraš će kupiti [illegible] šta drug[illegible] gumeno-plastični Sinclair Sp[illegible] K Razlog za to je vrlo [illegible] kompjutar (mnogi ga nazivaju [illegible] tako nesavršen a njegova koncepcija [illegible] puna rupa da je očigled[illegible] navrat-nanos Znači treba [illegible] od ružnog pečela napravi[illegible] labuda (jer je to u spektrumovom [illegible] nemoguća) ono bar nes[illegible]

Ako imate dotičn[illegible] računar [illegible] cu hardvaraša u seb[illegible] ugradili resat taster [illegible] lo će mu drastično p[illegible] Ines[illegible] tastatura će p[illegible] trum manje grafe [illegible] kompjuter Time ste [illegible] nedostatke vešeg ljub[illegible] česopisi koji iz broja u broj [illegible] raznih hardverskih pr[illegible] ra Potrabni su vam još Kempster [illegible] A/D konvertor EPROM program[illegible] EPROM disk RAM disk RS [illegible] fejs [illegible] Ako imate u planu [illegible] strukcija (a one su vam neop[illegible] je sigurno ne najboljem ste pu[illegible] te svoj kompjuter! Čak iako sve [illegible] vite vrlo pažljivo znači bez pogrešn[illegible] pogrešno ugrađenih elemenata [illegible] ugrađanih najsprevnih elemenata [illegible] preopteretićete stabilizator napona mikroprocsaor

Spektrumov stabilizator nap[illegible] de se optereti strujom manjom od 1 A pod uslovom da je pričvršćen na propisan [illegible] njak Ako su priključani interfejs [illegible] mikrodrajva on je već maksimalno [illegible] can i prema tome treba biti op[illegible] priključenjem ostalih periferija Ov[illegible] blem se lako rešava tako što ćete [illegible] jedan stabilizator 7805 [illegible] dograđena sklopova Naravno [illegible] treba preterivati jer su sada dodatn[illegible] rećani ispravljac i transform[illegible]

Problem preopterećenja mik[illegible] je nešto složeniji ali rešiv Radi se [illegible] ćem izlazi na magistralu adresa [illegible] i kontrolni izlazi procesora Z 80 [illegible] opteretiti sa maksimalno 1 8 mA [illegible] elektronskih elemenata priključen[illegible] izlaza ih opterećuje [illegible] zav[illegible] elamenta Sledeća tabela to poka[illegible]

[illegible] standardna TTL koje prave opas[illegible] esor Upravo to je razlog što se [illegible] instrukcijama komp[illegible] tera za k[illegible] koje ste sreli koriste LS TT[illegible] kola koja [illegible] terećuju 10 puta manje [illegible] je ovim kolima potreb[illegible] manja str[illegible] napajanja Znači postoje dva [illegible] spravite grešnog ser Klajva [illegible] nap[illegible] te nešto što bi [illegible] na kompletan kompjuter Jedan je da strog[illegible] svak[illegible] kolu priklju[illegible] čenom [illegible] svakom A kojim ga [illegible] opterećuje [illegible] je da osnazi [illegible] za to predvi[illegible] po shemi sa slike

[illegible] dataka upotre[illegible] tri stajt draj[illegible] je konstuisano [illegible] podataka odra[illegible] na logičkoj [illegible] ka magi[illegible] je suprotan [illegible] eno i ze adresnu liniju među to nije tipična namene 74 LS 245 jer je ovde komunikacija jednosmerna od procesora ke magistrali Za tri-stejt kontrolne izlaze *MREQ IORQ RD WR, RFSH* koristi se šsatobitni tri-stejt drajver 74 LS 367 Preostale tri kontrolna izlaze *M1 BUSAK HALT* su povezani sa 74 LS 04 Ovo kolo je upotrebljeno jer nam je potreban invertovan BUSAK signal On se neime koristi ze upravljenja tri-stajt drajverime i to ne sledeći način ked periferijski uređej pošelje zahtev za oslobeđenje magistrala edresa i podetaka *(BUSRQ* na nulu) procesor završi ciklus koji je obavljeo, svoje izleze ne megistrale postavi u stanje visoke impedense i izlaz *BUSAK* (busacknowledge — potvrda da su magistrale slobodne) dovede na nivo logičke nule Taj signel se invertuje i drejvere dovodi takođe u stenje visoke impedense Tako se ovi ponašaju identično procesoru i oslobađeju megistrale koje su napetli periferijskom uređeju koji je uputio zahtev

Miroslav Milošević

računari su uvek aktuelni

U prethodnim brojevima:

KORISTITE NARUDŽBENICU NA SLEDEĆOJ STRANI

Računari 1 — RASPRODATO

- Upotraba računara RAČUNARI NA STO NAČINA
- IZBOR računara JABUKA ISKUŠENJA
- Najpopularniji modeli računara ZVEZDE KOJE NE TAMNE
- Periferijska oprema VRATA U SVET — štampači disk jedinice ploteri palice za igru
- Hardverski dodaci SITNICE KOJE ŽIVOT ZNAČE
- Komercijalni programi DŽUNGLA U RAČUNARU
- Igra HAJDE DA SE IGRAMO
- Specijalizovani časopisi i knjiga HAKERSKA LEKTIRA
- Napravi i ti računar GALAKSIJA
- Umetnost programiranja MALE TAJNE VELIKIH MAJSTORA PROGRAMIRANJA
- Programiranja za početnika GOVORITE LI BEJZIK?
- Mali rečnik bajzika

Računari 2 — RASPRODATO

- Računari u Jugoslaviji NI GDLUB NA GRANI NI VRABAC U RUCI
- Računari u svetu NA ZAPADU NIŠTA NOVO
- Evolucija mozga od dinosaurusa do kompjutera U SUSRET SUPERINTELIGENCIJI
- Računari iz drugog ugla NEMOJTE PLAKATI ZA EPLOM
- Računari u izlogu ČEKAJUĆI OL-a ELEKTRON BBC COMMODORE 64
- Periferijska oprema ŠTAMPAČI U AKCIJI
- Računari i obrazovanja GALAKSIJA U ŠKOLI
- Programiranja za početnika MOJ PRVI PROGRAM
- Majstorija na računaru GALAKSIJA BEZ TAJNI
- Igra na računaru ŠKOLA AKCIONIH IGARA
- Računari u domaćoj radionosti PROGRAMATOR EPROM-a
- Galaksija 48K MEMORIJA ZA GALAKSIJU'
- Umetnost programiranja SVI SPEKTRUMOVI'' RESTARTI

Računari 3

- Računari u izlogu AMSTRAD CPC 464 ELEKTRON SPEKTRUM PLUS
- Periferijska oprema DISK JEDINICE
- Programi koja treba imati NEKI BOLJI BEJZICI
- Majstorija na računaru Spektrum — HALO DA LI JE TO MAŠINACI BBC/Elektron — ŠTA EJKORN NIJE REKAO?
- Računar galaksija ROM 2
- Katalog najboljih igara svih vremena za spektrum PEDESET VELIČANSTVENIH
- Generisanja naših slova u printerima epson ĆIRILICA NA EPSONU
- Škola avanturističkih igara UKLETI DVORAC I DRUGE BAJKE
- Simulacija letenja LETAČI BEZ DIPLOME
- Računari u domaćoj radinosti PROGRAMATOR EPROM-a

Računari 4

- Računari kroz istoriju DESET LUDIH GODINA
- Računari i njihova zamka IMATI ILI UMRETI
- Naš test ATARI 800 XL
- Uporedni test AMSTRAD PROTIV KOMODORA
- Računari u akciji DISKETNA VEZA
- Računar u poslovnoj primeni MAŠINA SA BEZBROJ LICA
- Računari u školi HILJADU ZATO ZA RAČUNARE
- Umetnost programiranja MODERNE PROGRAMERSKE TEHNIKE
- Majstorija na računaru ZX Spektrum — NOVE NAREDBE I FUNKCIJSKI TASTERI
- Računari na brzinskom lapitu SPRINTERI U KUĆICI PUŽA
- Nova klasa mikroprocesora DŽINOVI NA GLAVI ČIODE
- Katalog najboljih igara za komodor' PEDESET NAJBOLJIH
- Programiranja na bajziku NOVE AZBUKE NA KOMODORU PROMENLJIVE BEZ TAJNI
- Škola sistemskog softvera PUT U SREDIŠTE ROM-a
- Škola simulacija letenja DOBAR LET ELEKTRONSKA PTICO
- Računari u domaćoj radionosti GENERATOR TONA

Računari 5

- Računari u izlogu OVDE OJ DŽINOVA
- Periferijska oprema grafička tabla — RADOST CRTANJA NEKI NOVI EPSONI
- Računari i obrada teksta PISANJE BEZ MUKE
- Sedam načina komuniciranja sa kompjuterom TAKO ĆE GOVORITI RAČUNARI
- Računari u akciji STROGO KONTROLISANI DISKOVI
- Škola sistemskog programiranja RASPODELA MEMORIJE
- UMETNOST PROGRAMIRANJA Spektrum — INTERAPTI U BEJZIKU
- Majstorija na računaru Nove naredbe na spektrumu — NA SVOJU SLIKU I PRILIKU
- Programiranja u bajziku SPAJTOVI NA „KOMODORU' U svetu slučajnih brojeva ZEC IZ ŠEŠIRA
- Računari u domaćoj radionosti PRIČA O FINOJ GRAFICI

Računari 6

- Računari u izlogu Enterprajz — PREDUZIMLJIVOST SA ZADRŠKOM
- Promašaj na japanski način RAČUNARI IZLAZEĆEG SUNCA
- Periferijska oprema MONITORI
- Programi na ispitu NAKERSKI ZAKONIK
- Programi koja treba imati SLIKARI BEZ KIČICE
- Računari i obrada teksta PERO OD OSAM BITA
- Majstorija na računaru komodor 64 — NOVE NAREDBE NA NOVI NAČIN
- Umetnost programiranja spektrum — NOVA KRUNA ZA STAROG KRALJA
- Put u središte ROM-a VELIKA VIDEO PREDSTAVA
- Majstorija na računaru spektrum — EKRAN POD PRESOM
- Programiranja u bajziku HAMLET U RAČUNARU — Kako kompjuteri donose odluke
- Umetnost programiranja SVIRKA NA KOMODORU
- Računari i igra BESMRTNOST I KAKO JE STEĆI
- RAČUNARI u domaćoj radinosti Spektrum — ROM OD SEDAM MILJA Galaksija — FINI HARDVER ZA FINU GRAFIKU

Računari 7

- Naš test KANON PROTIV EPSONA'
- Računari u izlogu Komodor 128 — O KOMODORU SVE NAJLEPŠE
- Periferijska oprema SVETLOSNA PERA
- Umetnost programiranja HAKER NA USIJANOM LIMENOM KROVU
- Računari u akciji BAZE PODATAKA
- Programi koje treba imati NE DAJ SE INESI
- Računari u poslovnoj primeni MALI RAČUNARI U VELIKOJ PRIVREDI
- Struktura memorija i potprogrami iz ROM-a — SVE SPEKTRUMOVE RUTINE
- Test sa zadrškom PREMA SVECU I DRAJV
- Majstorija na računaru Komunikacija sa periferijskim uređajima PUTOVANJE U PROVINCIJE
- Programiranja u bajziku FUNKCIJE POTPROGRAMI PROCEDURE
- Grafika na računaru KAKO TO RADI AMSTRAD'
- Programiranja na mašincu Spektrum — KODOVI IZ RUKAVA
- Put u središte ROM-a LIČNE STVARI BEJZIKA
- Računari u domaćoj radinosti ROM OD SEDAM MILJA (2) BRISAČ EPROMA



1



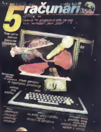
2



3



4



5



6

NARUDŽBENICA

Galaksija Bulevar vojvode Mišića 17, 11000 Beograd

Ovim neopozivo naručujem sledeća specijalna izdanja časopisa Galaksija 3—4—5 po ceni od 200 dinara po primerku odnosno 7—8 po ceni od 250 dinara (zaokružiti odgovarajuće brojeve) iznos od ukupno ______ dinara uplatiću poštaru prilikom preuzimanja pošiljke — POUZEĆEM

Ime i prezime ______

Ulica i broj ______

Broj pošte i mesto ______

______
Datum

______
Potpis



7




računari
9
AMIGA